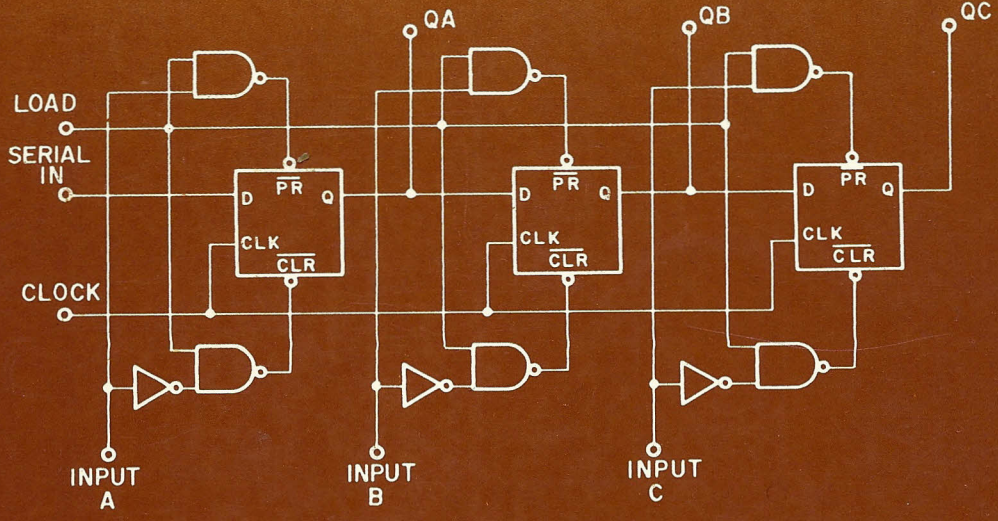
JOÃO BATISTA DE AZEVEDO JÚNIOR

TTL/CMOS

TEORIA E APLICAÇÃO EM CIRCUITOS DIGITAIS

4.ª EDIÇÃO



VOLUME 1

ÉRICA

CIP-Brasil. Catalogação-na-Publicação
Câmara Brasileira do Livro, SP

A986t
v.1-2

Azevedo Júnior, João Batista de, 1956-
TTL/CMOS : teoria e aplicação em circuitos digitais / João Batista de Azevedo Júnior. -- São Paulo : Érica, 1984.

Bibliografia.
1. Circuitos integrados 2. Eletrônica digital
I. Título.

84-0494

CDD-621.38153
-621.3815
-621.38173

Índices para catálogo sistemático:

1. Circuitos digitais : Engenharia eletrônica 621.38153
2. Circuitos integrados : Microeletrônica 621.38173
3. Eletrônica digital : Engenharia 621.3815

JOÃO BATISTA DE AZEVEDO JÚNIOR

TTL/CMOS

TEORIA E APLICAÇÃO EM CIRCUITOS DIGITAIS

ANO: 1993 92 91 90
EDIÇÃO: 10 9 8 7 6 5 4

LIVROS ÉRICA EDITORA LTDA.

Nenhuma parte desta publicação poderá ser reproduzida, guardada pelo sistema "retrieval" ou transmitida de qualquer modo ou por qualquer outro meio, seja este eletrônico, mecânico, de fotocópia, de gravação, ou outros, sem prévia autorização por escrito desta EDITORA.

PRODUÇÃO EDITORIAL: Paulo Roberto Alves

REVISÃO: Guaraciaba Micheletti

COMPOSIÇÃO DOS TEXTOS: Gilza Álvaro da Silva
Roseli Barbaresco de Oliveira

DESENHOS: Dernival Cordeiro da Silva

LIVROS ÉRICA EDITORA LTDA.
Rua Jarinu, 594 - Tatuapé - São Paulo
Fone: 294-8686 - C.G.C. 50.268.838/0001-39
Caixa Postal 15.617

PREFÁCIO

Este livro tem como finalidade expor de maneira clara e objetiva, fartamente ilustrada com aplicações práticas, os elementos básicos dos circuitos digitais e sua implementação com circuitos integrados.

Considerando a disponibilidade de componentes no mercado nacional, a discussão volta-se para CI's TTL da série 74 e CMOS da série 4000, sem dúvida os mais comumente encontráveis na atualidade.

Em todos os capítulos onde se trata da implementação de algum elemento (contadores, decodificadores, etc), procurou-se apresentar diversos CI's de ambas as séries que podem ser usados para realizar a função discutida, juntamente com suas características, e ilustrar circuitos construídos com esses elementos. Procurou-se, também, salientar as considerações práticas envolvidas, apontando os cuidados que devem ser tomados no uso de CI's, os possíveis problemas que podem ocorrer e o que se deve fazer para evitá-los.

O material e sua apresentação permitem que a obra seja utilizada como livro-texto para a cadeira de Eletrônica Digital a nível de curso técnico, ou em um curso introdutório a nível de 3º grau.

O autor deseja expressar seus agradecimentos a todos que contribuíram para a realização desta obra e em especial ao amigo Waldir João Sandrini, cujas críticas e incentivo foram muito valiosos.

O autor.

SUMÁRIO

VOLUME I

CAPÍTULO 1 - SISTEMAS DE NUMERAÇÃO 11

1.1 - Introdução..... 11
1.2 - Mudanças de Base Qualquer para Base 10..... 12
1.3 - Mudança da Base 10 para Base "b" Qualquer.. 13
1.4 - Conversão entre Bases 2^n (n = 1,2,3,...)... 16

CAPÍTULO 2 - ÁLGEBRA DE BOOLE 19

2.1 - Introdução..... 19
2.2 - Variáveis Lógicas..... 19
2.3 - Lógica Positiva e Lógica Negativa..... 20
2.4 - Operações Lógicas..... 20

CAPÍTULO 3 - FAMÍLIAS LÓGICAS 45

3.1 - Introdução..... 45
3.2 - Famílias de Circuitos Integrados..... 45
3.3 - Família TTL..... 46
3.4 - Família CMOS..... 72
3.5 - Interface TTL/C-MOS e C-MOS/TTL..... 84
3.6 - Comparação entre TTL e CMOS..... 86

CAPÍTULO 4 - PORTAS LÓGICAS E CIRCUITOS COMBINACIONAIS . 87

4.1 - Introdução..... 87
4.2 - Portas Lógicas da Família TTL..... 87
4.3 - Portas Lógicas da Família C-MOS..... 92
4.4 - Análise de Circuitos Combinacionais..... 99
4.5 - Projeto de Circuitos Combinacionais..... 106
4.6 - Exemplos de Aplicação..... 134

CAPÍTULO 5 - FLIP-FLOPS 143

5.1 - Introdução..... 143
5.2 - FLIP-FLOP com Portas Lógicas..... 144
5.3 - FLIP-FLOP RS MASTER-SLAVE..... 148
5.4 - FLIP-FLOP JK MASTER-SLAVE..... 153
5.5 - FLIP-FLOPS D e T..... 157
5.6 - Obtenção de um Tipo de FF a partir de outro 158
5.7 - FLIP-FLOP da Família TTL..... 162
5.8 - FLIP-FLOP da Família CMOS..... 171

CAPÍTULO 6 - **MULTIVIBRADORES ASTÁVEIS, MONOESTÁVEIS E DISPARADORES SCHMITT**..... 177

6.1 - Introdução..... 177
6.2 - Osciladores RC com Circuitos Integrados.... 177
6.3 - Osciladores a Cristal..... 184
6.4 - Temporizador 555..... 187
6.5 - Multivibradores Monoestáveis..... 192
6.6 - Disparadores SCHMITT..... 199
6.7 - Distribuição de CLOCKS..... 203
6.8 - Monoestáveis e Disparadores SCHMITT da Família TTL..... 205
6.9 - Monoestáveis e Disparadores SCHMITT da Família CMOS..... 208

SUMÁRIO

VOLUME II

CAPÍTULO 7 - CONTADORES DIGITAIS 221

7.1 - Introdução..... 221
7.2 - Contadores Assíncronos..... 222
7.3 - Contadores Síncronos..... 232
7.4 - Contadores Síncronos de Seqüência Qualquer. 235
7.5 - Contadores da Família TTL..... 243
7.6 - Contadores da Família C-MOS..... 253
7.7 - Aplicações de Contadores MSI..... 259

CAPÍTULO 8 - REGISTRADORES DE DESLOCAMENTO 267

8.1 - Introdução..... 267
8.2 - Tipos de SR's..... 268
8.3 - Registradores de Deslocamento da Família TTL.. 270
8.4 - Registradores de Deslocamento da Família CMOS. 275
8.5 - Aplicações dos Registradores de Deslocamento 278

CAPÍTULO 9 - CÓDIGOS BINÁRIOS E DECODIFICADORES 299

9.1 - Introdução..... 299
9.2 - Códigos Binários..... 299
9.3 - Decodificadores..... 304
9.4 - Decodificadores, Multiplexadores e Demultiplexadores da Família TTL..... 312
9.5 - Decodificadores, Multiplexadores e Demultiplexadores da Família CMOS..... 321
9.6 - Aplicações de Decodificadores, Multiplexadores e Demultiplexadores..... 328

CAPÍTULO 10 - ELEMENTOS ARITMÉTICOS 337

10.1 - Introdução..... 337
10.2 - Adição de Números Binários..... 337

10.3 - Subtração de Números Binários..... 345
10.4 - Representação de Números Binários em Sistemas Digitais..... 348
10.5 - Geração e Verificação de Paridade..... 351
10.6 - Elementos Aritméticos da Família TTL..... 353
10.7 - Elementos Aritméticos da Família CMOS..... 360
10.8 - Aplicações..... 362

CAPÍTULO 11 - MEMÓRIAS SEMICONDUTORAS 365

11.1 - Introdução..... 365
11.2 - Características Gerais..... 366
11.3 - Memórias ROM..... 367
11.4 - Memórias RAM..... 373
11.5 - Aplicações..... 378

CAPÍTULO 12 - CONVERSORES D/A E A/D 387

12.1 - Introdução..... 387
12.2 - Conceitos Básicos..... 387
12.3 - Conversores Digital-Analógico..... 393
12.4 - Conversores Analógico-Digital..... 397
12.5 - Parâmetros dos Conversores D/A e A/D..... 403

CAPÍTULO 1 - SISTEMAS DE NUMERAÇÃO

1.1 - Introdução

Em nosso dia a dia, estamos habituados a trabalhar com números decimais ou de base 10. Consideremos o significado do número 1983. O primeiro algarismo (1) representa um milhar, o segundo (9) nove centenas, o terceiro (8) oito dezenas e o quarto (3) três unidades, ou seja,

$$1983 = 1 \times 1000 + 9 \times 100 + 8 \times 10 + 3 \times 1$$
$$= 1 \times 10^3 + 9 \times 10^2 + 8 \times 10^1 + 3 \times 10^0$$

Vemos que cada algarismo possui um valor absoluto(um, nove, oito, três) e outro relativo, que decorre de sua posição. Cada posição corresponde a uma potência de 10, a saber 10^0 (unidades), 10^1 (dezenas), 10^2 (centenas), etc.

Este sistema de numeração decimal ou de base 10 é comumente empregado. Nada impede, entretanto, que usemos um sistema de numeração de base "b" qualquer (b inteiro e maior que 1). Na Eletrônica Digital, interessa-nos conhecer os sistemas de numeração de bases, 2, 8 e 16.

A figura 1.1 mostra algumas bases possíveis e os algarismos que cada uma utiliza.

SISTEMA	BASE	ALGARISMOS
Binário	2	0,1
Ternário	3	0,1,2
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Duodecimal	12	0,1,2,3,4,5,6,7,8,9,A,B
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Fig. 1.1 - Sistemas de Numeração.

1.2 - Mudanças de Base Qualquer para Base 10

Consideremos um número n, escrito em uma base "b" qualquer.

$$n = X_n\ X_{n-1}\ X_{n-2} \ldots X_0,\ Y_1\ Y_2 \ldots Y_n$$

onde

X_n, X_{n-1}, ..., X_0 - algarismos da parte inteira do número.

$Y_1\ Y_2$, ..., Y_n - algarismos da parte fracionária do número.

Para encontrar o equivalente no sistema decimal basta multiplicar cada algarismo da parte inteira, X_n ($n = 0,1,2\ldots$), por b^n e cada algarismo da parte fracionária, Y_n ($n = 1,2\ldots$), por b^{-n} e somar os resultados.

Exemplo 1.1:

Converter os números escritos nas bases indicadas para a base 10:

a) $101_{(2)}$ b) $211_{(8)}$ c) $AF4_{(16)}$

d) $1001,11_{(2)}$ e) $76,31_{(8)}$ f) $D21,FC_{(16)}$

Solução:

a) $101_{(2)} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5_{(10)}$

b) $211_{(8)} = 2 \times 8^2 + 1 \times 8^1 + 1 \times 8^0 = 128 + 8 + 1 = 137_{(10)}$

c) $AF4_{(16)} = 10 \times 16^2 + 15 \times 16^1 + 4 \times 16^0 = 2560 + 240 + 4 = 2804_{(10)}$

d) $1001,11_{(2)} = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
$= 8 + 0 + 0 + 1 + 0,5 + 0,25 = 9,75_{(10)}$

e) $76,31_{(8)} = 7 \times 8^1 + 6 \times 8^0 + 3 \times 8^{-1} + 1 \times 8^{-2} = 62,390625_{(10)}$

f) $021,FC_{(16)} = 13 \times 16^2 + 2 \times 16^1 + 1 \times 16^0 + 15 \times 16^{-1} + 12 \times 16^{-2}$

$= 3361,984375_{(10)}$

1.3 - Mudança da Base 10 para Base "b" Qualquer

Vejamos primeiro o caso de números inteiros.

Dado um número inteiro escrito na base 10, para obter seu equivalente em uma base "b" qualquer, divide-se o número por "b" tantas vezes quantas necessárias para que o quociente da divisão seja menor que "b"; o último quociente e os restos das divisões sucessivas, tomados na ordem inversa, correspondem ao número na base "b".

Exemplo 1.2:

Converter os números inteiros escritos na base 10 para as bases indicadas.

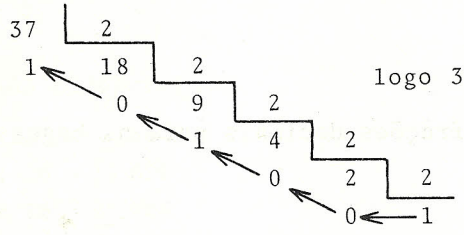
a) 37 para base 2

b) 936 para base 8

c) 10523 para base 16

Solução:

a) Aplicando a regra explicada

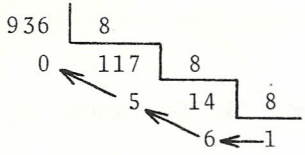


logo $37_{(10)} = 100101_{(2)}$

Um algoritmo muito conveniente é mostrado abaixo:

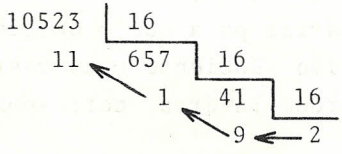
37	18	9	4	2	1
1	0	1	0	0	1

b) Aplicando a regra



logo $936_{(10)} = 1650_{(8)}$

c) Aplicando a regra



logo $10523_{(10)} = 291B_{(16)}$, onde se considerou que $11_{(10)} = B_{(16)}$

Quando se deseja converter uma fração decimal para outra base, utiliza-se o seguinte procedimento: multiplica-se a fração pela base e toma-se a parte inteira do produto como primeiro algarismo do número desejado; a parte fracionária do produto anterior é novamente multiplicada pela base e a parte inteira deste novo produto é o segundo algarismo do número desejado; o processo é repetido até se obter a precisão requerida ou a parte fracionária de um dos produtos ser igual a 0.

Exemplo 1.3:

Converter as frações decimais para as bases indicadas.

a) 0,875 para base 2

b) 0,746 para base 2

c) 0567 para base 8

d) 0,354 para base 16

Solução:

a) Aplicando a regra:

$0,875 \times 2 = \underline{1},75$
$0,75 \times 2 = \underline{1},50$
$0,50 \times 2 = \underline{1},00$

logo $0,875_{(10)} = 0,111_{(2)}$

b) Aplicando a regra:

$0,746 \times 2 = \underline{1},492$
$0,492 \times 2 = \underline{0},984$
$0,984 \times 2 = \underline{1},986$
$0,986 \times 2 = \underline{1},972$
$0,972 \times 2 = \underline{1},944$

logo $0,746_{(10)} = 0,10111_{(2)}$ com 5 casas de aproximação (erro menor que 2^{-5}).

c) Aplicando a regra:

$0,567 \times 8 = \underline{4},536$
$0,536 \times 8 = \underline{4},288$
$0,288 \times 8 = \underline{2},304$
$0,304 \times 8 = \underline{2},432$
$0,432 \times 8 = \underline{3},456$

logo $0,567_{(10)} = 0,44223_{(8)}$ com 5 casas de aproximação (erro menor que 8^{-5}).

d) Aplicando a regra:

$0,345 \times 16 = \underline{5},664$
$0,664 \times 16 = \underline{10},624$
$0,624 \times 16 = \underline{9},984$
$0,984 \times 16 = \underline{15},744$
$0,744 \times 16 = \underline{11},904$

logo $0,345_{(10)} = 0,5A9FB_{(16)}$ com 5 casas de aproximação (erro menor que 16^{-5}).

Quando o número na base 10 possui parte inteira e parte fracionária, convertemos separadamente cada uma das partes, pelos métodos indicados, unindo depois os resultados.

1.4 - Conversão Entre Bases 2^n (n = 1,2,3,...)

É útil fazer a conversão de octal ou hexadecimal para binário, ou vice-versa, sem utilizar a conversão para decimal como passo intermediário.

Dado um número escrito numa base 2^n (n = 3 para octal, n = 4 para hexadecimal) pode-se converter diretamente cada algarismo para o binário equivalente com n bits (bit = BINARY DIGIT).

Exemplo 1.4:

Converter para binário os números escritos nas bases indicadas

a) $346_{(8)}$ b) $531_{(16)}$ c) $532_{(8)}$ d) $ABC_{(16)}$

Solução:

a) Convertendo diretamente cada algarismo para o binário de 3 bits correspondente

$$346_{(8)} = 011100110_{(2)}$$

b) Convertendo diretamente cada algarismo para o binário de 4 bits correspondente

$$531_{(16)} = 0101\ 0011\ 0001_{(2)}$$

c) $532_{(8)} = 101\ 011\ 010_{(2)}$

d) $ABC_{(16)} = 1010\ 1011\ 1100_{(2)}$

Dado um número em binário, para convertê-lo a uma base 2^n, divide-se o número em grupos de n bits da direita para a esquerda, completando o grupo mais a esquerda com zeros, se necessário, e converte-se cada grupo para a base desejada.

Exemplo 1.5:

Converter para as bases indicadas os números binários

(a) 101110 para octal

(b) 1100111101 para hexadecimal

Solução:

a) Dividindo o número em grupos de 3 bits a partir da direita

$$101110_{(2)} = 101\ 110 = 56_{(8)}$$

b) Dividindo o número em grupos de 4 bits a partir da direita

$$1100111101_{(2)} = 0011\ 0011\ 1101 = 33D_{(16)}$$

CAPÍTULO 2 - ÁLGEBRA DE BOOLE

2.1 - Introdução

Na Eletrônica Digital trabalhamos com grandezas que assumem apenas dois valores distintos, ou seja, grandezas binárias. A ferramenta matemática utilizada no tratamento deste tipo de grandezas chama-se Álgebra de Boole (ou booleana) e foi desenvolvida em meados do século XIX pelo matemático inglês George Boole (1815-1864).

Neste capítulo, faremos uma exposição breve e informal da álgebra booleana, tendo em vista sua aplicação no projeto e desenvolvimento de sistemas digitais.

2.2 - Variáveis Lógicas

Variáveis lógicas são aquelas que somente podem assumir dois estados distintos. Estes dois estados são chamados:

- Ø e 1;
- LO e HI;
- VERDADEIRO e FALSO;
- ON e OFF;

e assim por diante. Neste livro, empregaremos os símbolos "Ø" e "1" para representar os dois estados lógicos distintos.

Considere o circuito da fig. 2.1 para ilustrar o conceito da variável lógica. Veja que a tensão no ponto A será igual a:

- 5 Volts, se a chave S estiver aberta;
- Ø Volts, se a chave S estiver fechada.

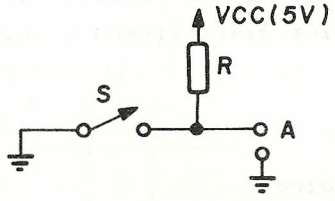


Fig. 2.1 - Exemplo de variável lógica.

Assim sendo, a tensão no ponto A é uma variável lógica, pois somente poderá tomar dois valores distintos. Neste livro, representaremos as variáveis lógicas por letras maiúsculas (A, B, C, Y, etc).

2.3 - Lógica Positiva e Lógica Negativa

No circuito da fig. 2.1, vimos que a tensão no ponto A pode ser igual a 5 V ou Ø V, conforme a posição da chave S. Temos completa liberdade para escolher qual dos valores de tensão chamaremos de "1" (ou HI, ou VERDADEIRO, ou ON, etc.)e qual chamaremos de "Ø" (ou LO, ou FALSO, ou OFF, etc). Da escolha que fizermos dependerá o fato de passarmos a utilizar lógica positiva ou lógica negativa.

Definimos lógica positiva como aquela na qual o estado lógico "1" é intuitivamente superior ao estado lógico "Ø".De acordo com este conceito, podemos estabelecer que:

5 Volts = estado lógico "1" }
Ø Volts = estado lógico "Ø" } lógica positiva

5 Volts = estado lógico "Ø" }
Ø Volts = estado lógico "1" } lógica negativa

Nos itens seguintes definiremos uma série de operações lógicas. A definição destas operações é completamente independente do tipo de lógica (positiva ou negativa) utilizado,mas *a função realizada por um circuito lógico depende da convenção adotada*. Nos exemplos que são apresentados após a definição de cada uma das operações lógicas usaremos sempre a lógica positiva; num item posterior discutiremos o que ocorre quando se alteram as convenções.

2.4 - Operações Lógicas

Você está familiarizado e trabalha há muito tempo com

funções algébricas. Por exemplo, na função

$$y = 3x + 2z - t$$

você sabe que x, z e t são as variáveis independentes e que y é a variável dependente, pois seu valor depende dos valores de x, z e t; as variáveis independentes estão relacionadas pelas operações de soma, subtração, multiplicação, divisão, ou outras, cujas regras foram definidas em algum ponto anterior.

Vamos conhecer agora outros tipos de funções, que são as funções lógicas. Uma função lógica tem a forma:

$$Y = f(A, B, C, ...)$$

onde Y, A, B, C, etc, são *variáveis lógicas*, possuindo em consequência somente dois estados possíveis. Y é a variável dependente e A, B, C, etc. são as variáveis independentes; Y poderia representar a saída de um circuito digital cujas entradas fossem A, B, C, etc. As variáveis independentes estão relacionadas através de *operações lógicas*, que iremos definir nos itens seguintes.

2.4.1 - Operação NÃO

A operação "NÃO" (em inglês: NOT) é a mais simples das operações lógicas. Ela simplesmente associa a uma variável lógica o seu inverso. Se A é uma variável lógica, seu inverso é representado por $\overline{A}$ (leia-se não-A ou A-barra). A figura 2.2 mostra a tabela da operação "NÃO" ou "Negação".

A	Y
Ø	1
1	Ø

$Y = \overline{A}$

Fig. 2.2 - Tabela da operação NÃO.

A tabela mostrada na figura 2.2 chama-se tabela da verdade (em inglês: TRUTH TABLE) e é uma forma extremamente útil de representar funções lógicas. A coluna da esquerda mostra os valores que A pode assumir e a coluna da direita os valores resultantes para a variável Y. Como neste caso temos uma única variável independente ou de entrada, é claro que a tabela da verdade possui duas linhas, uma vez que A somente pode ser "Ø" ou "1".

Se tivéssemos duas variáveis de entrada, A e B, a tabela da verdade teria quatro linhas, porque A e B poderiam se apresentar nas combinações ØØ, Ø1, 1Ø e 11. Para três variáveis, A, B e C, seriam oito as linhas da tabela, porque as combinações possíveis são ØØØ, ØØ1, Ø1Ø, Ø11, 1ØØ, 1Ø1, 11Ø e 111. É fácil concluir que para n variáveis de entrada a tabela da verdade possui 2^n linhas, correspondendo às 2^n combinações possíveis entre as variáveis. A figura 2.3 ilustra as tabelas da verdade para 2,3 e 4 variáveis.

A	B	Y
Ø	Ø	
Ø	1	
1	Ø	
1	1	

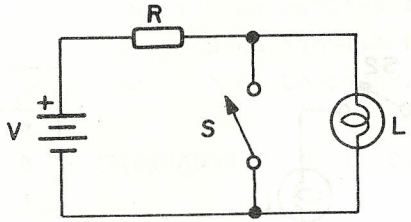
A	B	C	Y
Ø	Ø	Ø	
Ø	Ø	1	
Ø	1	Ø	
Ø	1	1	
1	Ø	Ø	
1	Ø	1	
1	1	Ø	
1	1	1	

A	B	C	D	Y
Ø	Ø	Ø	Ø	
Ø	Ø	Ø	1	
Ø	Ø	1	Ø	
Ø	Ø	1	1	
Ø	1	Ø	Ø	
Ø	1	Ø	1	
Ø	1	1	Ø	
Ø	1	1	1	
1	Ø	Ø	Ø	
1	Ø	Ø	1	
1	Ø	1	Ø	
1	Ø	1	1	
1	1	Ø	Ø	
1	1	Ø	1	
1	1	1	Ø	
1	1	1	1	

Fig. 2.3 - Tabelas da Verdade para Funções de 2, 3 e 4 variáveis lógicas.

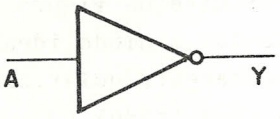
A figura 2.4 mostra um circuito elétrico que representa a operação "NÃO". Neste circuito, se a chave S está aberta a lâmpada L está acesa e se a chave S está fechada a lâmpada L está apagada. Na tabela da verdade mostrada na figura a variável de entrada é S, representando a posição da chave(S = "Ø" → chave aberta, S = "1" → chave fechada), e a variável de saída é L, representando o acendimento da lâmpada (L = "Ø" → lâmpada apagada, L = "1" → lâmpada acesa).

O circuito que realiza a operação "NÃO" é denominado inversor e representado usualmente pelo símbolo mostrado na figura 2.5.



S	L
Ø	1
1	Ø

Fig. 2.4 - Circuito Representativo da Operação NÃO.



$Y = \overline{A}$

Fig. 2.5 - Símbolo do circuito inversor .

2.4.2 - Operação E

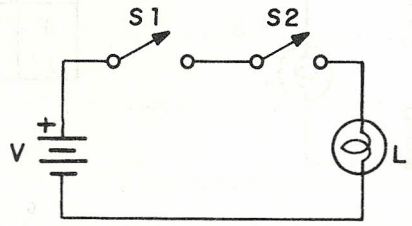
A operação "E" (em inglês: AND) é definida como aquela em que a variável de saída é igual a "1" se, e somente se, *todas* as variáveis de entrada forem iguais a "1". O símbolo usa

do para representar a operação "E" é um ponto. Assim, a expressão.

$$Y = A \cdot B$$

é lida como Y igual a A E B.

A figura 2.6 ilustra um circuito elétrico que representa a operação "E" para duas variáveis e sua tabela da verdade. As convenções utilizadas são as mesmas que para o circuito da figura 2.4 e você pode ver claramente que para que a lâmpada L esteja acesa (estado lógico "1") é necessário que as chaves S_1 e S_2 estejam fechadas (ambas no estado lógico "1").



S_1	S_2	L
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

Fig. 2.6 - Circuito Representativo e Tabela da Verdade para a operação "E".

Consideremos agora o circuito da figura 2.7, onde os diodos D_1 e D_2 são ideais (lembre-se: o diodo ideal se comporta como uma chave aberta quando reversamente polarizado e como uma chave fechada quando diretamente polarizado).

As variáveis de entrada são A e B (tensões nos pontos A e B, respectivamente) e a variável de saída é Y (tensão no ponto Y). Se o ponto A é aterrado (tensão no ponto A igual a Ø Volts), D_1 conduz e a tensão no ponto Y fica igual a Ø Volts. Se o ponto B é aterrado, D_2 conduz e a saída Y vai para Ø Volts, o mesmo ocorrendo se A e B são aterrados simultaneamente. Somente quando as tensões em A e B forem ambas iguais a 5 Volts, a saída Y ficará em 5 Volts, pois então nem D_1 nem D_2 estarão conduzindo.

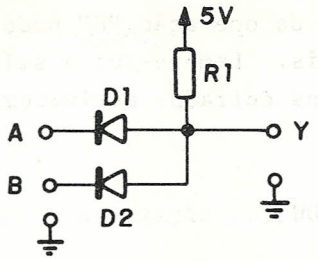


Fig. 2.7 - Circuito Lógico com Diodos

Os resultados desta análise estão resumidos sob forma de uma tabela na fig. 2.8. Esta tabela chama-se *tabela de estados* e é análoga a uma tabela da verdade, com a diferença que até o momento não decidimos que valor de tensão representará o estado lógico "Ø" e que valor representará o estado lógico "1". Convencionando que 5 Volts corresponde ao estado lógico "1" e que Ø Volts corresponde ao estado lógico "Ø", ou seja, utilizando lógica positiva, a tabela de estados se transforma na tabela da verdade também mostrada na fig. 2.8. Assim, podemos afirmar que o circuito da fig. 2.7 realiza a operação"E" com as variáveis de entrada, para lógica positiva.

A	B	Y
ØV	ØV	ØV
ØV	5V	ØV
5V	ØV	ØV
5V	5V	5V

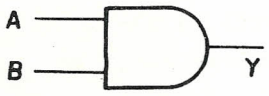
=>

A	B	Y
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

Y = A.B

Fig. 2.8 - Tabela de Estados e Tabela da Verdade para o Circuito da fig. 2.7 com Lógica Positiva.

O circuito que realiza a operação "E" denomina-se porta (em inglês: GATE) "E". O símbolo utilizado para representar a porta "E" está ilustrado na figura 2.9



Y = A.B

Fig. 2.9 - Símbolo de uma porta "E" para duas entradas (A e B).

O conceito da operação "E" pode ser estendido para mais de duas variáveis. Lembre-se: a saída estará em "1" se, e somente se, todas as entradas estiverem em "1".

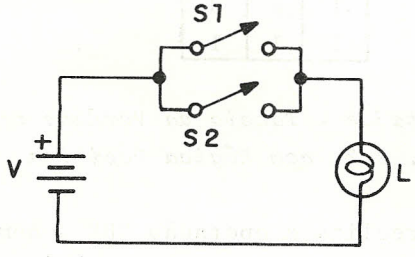
2.4.3. - Operação OU

A operação "OU" (em inglês: OR) é definida como aquela em que a variável de saída é igual a "1" se uma ou mais variáveis de entrada forem iguais a "1". Isto é equivalente a dizer que a saída será igual a "Ø" se, e somente se, *todas* as variáveis de entrada forem iguais a "Ø". O símbolo usado para representar a operação OU é uma cruz. Assim, a expressão:

$$Y = A + B$$

é lida como Y igual a A OU B.

A figura 2.10 mostra um circuito elétrico que representa a operação "OU" para duas variáveis e sua tabela da verdade. As convenções utilizadas são as mesmas que para o circuito da figura 2.4 e você pode ver claramente que a lâmpada L estará acesa (estado lógico "1") se a chave S_1 OU a chave S_2 (ou ambas) estiverem fechadas (estado lógico "1").

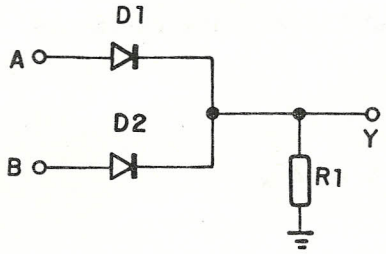


S_1	S_2	L
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

Fig. 2.10 - Circuito Representativo e Tabela da Verdade para a operação "OU".

Na figura 2.11, ilustra-se um circuito lógico com dio

dos ideais. Neste circuito vê-se que, se a entrada A ou a entrada B ou ambas estiverem em 5 Volts, a saída Y também irá para 5 Volts; Y será igual a Ø Volts se, e somente se, as duas entradas A e B estiverem em Ø Volts. A tabela de estados correspondente está também mostrada na figura 2.11, bem como a tabela da verdade para lógica positiva. Concluímos que o circuito realiza a operação OU com as variáveis de entrada, para lógica positiva.



A	B	Y
ØV	ØV	ØV
ØV	5V	5V
5V	ØV	5V
5V	5V	5V

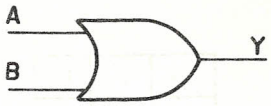
⟹

A	B	Y
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

Y = A + B

Fig. 2.11 - Circuito Lógico com Diodos, Tabela de Estados e Tabela da Verdade para Lógica Positiva.

A porta "OU" é representada usualmente pelo símbolo ilustrado na figura 2.12, que corresponde a uma porta de duas entradas.



Y = A + B

Fig. 2.12 - Símbolo de uma porta "OU" de duas entradas.

O conceito da operação "OU" pode ser facilmente estendido para mais de duas variáveis. Lembre-se: a saída estará em "1", se uma ou mais variáveis de entrada estiverem em "1".

2.4.4 - Operação NÃO - E

As operações "E", "OU" e "NÃO" são as operações básicas ou primitivas da álgebra booleana e todas as demais delas derivam. A seguir, estudaremos algumas operações mais complexas e que possuem enorme aplicação prática. Começaremos pela operação "NÃO-E" (em inglês: NAND).

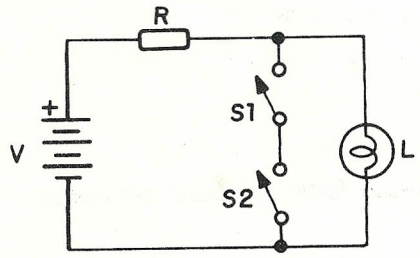
A operação "NÃO-E" é o inverso da operação "E". Assim, a saída Y será igual a "Ø" se, e somente se, todas as variáveis de entrada forem iguais a "1". A figura 2.13 mostra a tabela da verdade para a operação "NÃO-E" com duas variáveis.

A	B	Y
Ø	Ø	1
Ø	1	1
1	Ø	1
1	1	Ø

$$Y = \overline{A \cdot B}$$

Fig. 2.13 - Tabela da Verdade para a operação "NÃO-E".

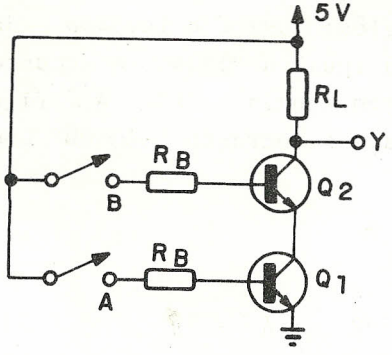
Na figura 2.14, mostra-se um circuito elétrico que representa a operação "NÃO-E" para duas variáveis. Vê-se, neste circuito, que a lâmpada L somente se apagará (estado lógico "Ø") quando ambas as chaves S_1 e S_2 estiverem fechadas (estado lógico "1").



S_1	S_2	L
Ø	Ø	1
Ø	1	1
1	Ø	1
1	1	Ø

Fig. 2.14 - Circuito Representativo da operação "NÃO-E" para duas variáveis.

A figura 2.15 ilustra um circuito lógico construído com transistores. Quando utilizado em circuitos lógicos, o transistor é operado como uma chave; isto quer dizer que ele estará ou na saturação ou no corte, não se utilizando a região ativa, onde a corrente de coletor "I_C" é proporcional à corrente de base "I_B". O circuito da figura 2.15 ilustra bem este fato. Quando uma das chaves (ou ambas) está aberta, o transistor correspondente não tem corrente de base e, por conseqüência, está cortado. Nesta situação a corrente através de Q_1 e Q_2 é muito pequena, sendo igual à corrente de fuga "I_{CEO}". A queda de tensão em R_L é mínima e a tensão no ponto Y é praticamente igual a 5 Volts. Se ambas as chaves estão fechadas, a corrente de base é suficiente para saturar os transistores, pois R_B é dimensionado para que isto ocorra. Na saturação a corrente de coletor é independente da corrente de base e limitada pelo valor de R_L; a tensão no ponto Y fica aproximadamente igual a Ø Volts, sendo a soma dos V_{CE}'s de saturação de Q_1 e Q_2. Estes resultados são resumidos na tabela de estados apresentada na figura, à qual se transforma na tabela da verdade mostrada, para lógica positiva. Então, concluímos que o circuito da figura 2.15 realiza a operação NÃO-E para duas variáveis.



A	B	Y
ØV	ØV	∿5V
ØV	5V	∿5V
5V	ØV	∿5V
5V	5V	∿ØV

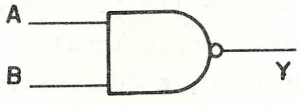
=>

A	B	Y
Ø	Ø	1
Ø	1	1
1	Ø	1
1	1	Ø

Fig. 2.15 - Circuito lógico com Transistores, Tabela de Estados e Tabela da Verdade para Lógica Positiva.

Dois pontos importantes podem ser destacados no exemplo anterior. Em primeiro lugar a utilização do transistor como elemento de comutação, comportando-se como uma chave aberta (corte) ou fechada (saturação). O outro ponto é menos óbvio. Imagine que a chave A esteja fechada e que fechamos a chave B. A passagem do estado de corte para o estado de saturação não é instantânea. Existe um pequeno atraso entre o fechamento da chave B e a transição do circuito de um estado para outro. Isto ocorre em qualquer circuito lógico e dedicaremos em outros capítulos muita atenção aos atrasos associados às portas e outros circuitos lógicos.

O circuito que executa a operação "NÃO-E" é representado usualmente pelo símbolo mostrado na figura 2.16.



$$Y = \overline{A \cdot B}$$

Fig. 2.16 - Símbolo de uma porta "NÃO-E" de duas entradas.

2.4.5 - Operação NÃO-OU

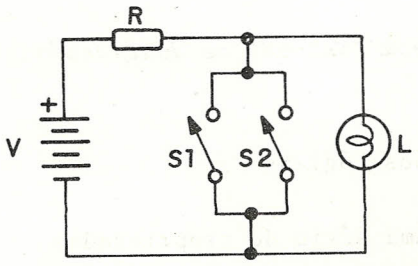
A operação "NÃO-OU" (em inglês: NOR) é o inverso da operação "OU". Assim, a saída Y será igual a "1" se, e somente se, *todas* as variáveis de entrada forem iguais a "Ø". A figura 2.17 mostra a tabela da verdade para a operação "NÃO-OU" com duas variáveis.

A	B	Y
Ø	Ø	1
Ø	1	Ø
1	Ø	Ø
1	1	Ø

$$Y = \overline{A + B}$$

Fig. 2.17 - Tabela da Verdade para a operação "NÃO-OU".

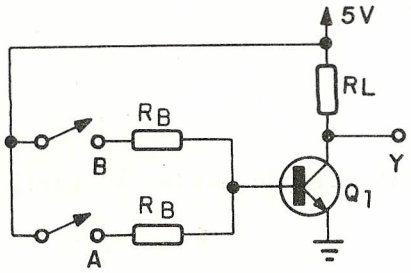
Um circuito elétrico representativo da operação "NÃO-OU" é ilustrado na figura 2.18. Neste circuito vemos que a lâmpada L somente estará acesa (estado lógico "1") quando as chaves S_1 e S_2 estiverem abertas (estado lógico "Ø").



S_1	S_2	L
Ø	Ø	1
Ø	1	Ø
1	Ø	Ø
1	1	Ø

Fig. 2.18 - Circuito Representativo da operação NÃO-OU para duas variáveis.

No circuito lógico mostrado na figura 2.19, vemos que se qualquer das chaves estiver fechada o transistor Q_1 irá para a saturação e a tensão de saída Y será igual ao V_{CE} de saturação de Q_1, ou seja, um valor próximo de Ø Volts. Somente quando as duas chaves estiverem abertas Q_1 estará cortado e a tensão de saída ficará aproximadamente igual a 5 Volts. A tabela de estados e a correspondente tabela da verdade para lógica positiva nos permitem afirmar que o circuito da fig. 2.19 realiza a função "NÃO-OU" para duas variáveis.



A	B	Y
ØV	ØV	∿5V
ØV	5V	∿ØV
5V	ØV	∿ØV
5V	5V	∿ØV

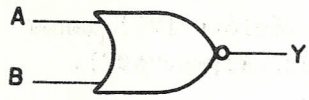
=>

A	B	Y
Ø	Ø	1
Ø	1	Ø
1	Ø	Ø
1	1	Ø

Fig. 2.19 - Circuito Lógico com Transistor, Tabela de Estados e Tabela da Verdade para Lógica Positiva.

A porta "NÃO-OU" é, geralmente, representada pelo sím

bolo mostrado na figura 2.20.



$$Y = \overline{A + B}$$

Fig. 2.20 - Símbolo de uma porta "NÃO-OU" de duas entradas.

2.4.6 - Propriedades das Operações Lógicas

Vamos agora examinar uma série de propriedades das operações lógicas, que serão de grande utilidade em itens posteriores. Faremos uma apresentação resumida destas propriedades, sem nos preocuparmos com as demonstrações, que podem ser feitas por você a título de exercício. Estas propriedades podem ser verificadas de maneira simples e imediata através das tabelas da verdade. Vejamos:

1) As operações "E" e "OU" são comutativas, ou seja

$$A \cdot B = B \cdot A$$
$$A + B = B + A$$

2) As operações "E" e "OU" são associativas, ou seja

$$A.(B.C) = (A.B).C$$
$$A+(B+C) = (A+B)+C$$

3) Teorema da involução

$$\overline{\overline{A}} = A$$

que quer dizer simplesmente que a dupla negação conduz à variável original (não-não-A é igual a A)

4) A operação "E" é distributiva em relação à operação "OU"

$$A.(B+C) = A.B + A.C$$

5) Outras propriedades

$A.A = A$

$A+A = A$

$A.\emptyset = \emptyset$

$A.1 = A$

$A+\emptyset = A$

$A+1 = 1$

$A.\overline{A} = \emptyset$

$A+\overline{A} = 1$

$A+A.B = A$

6) Teoremas de De Morgan:

1º) "A operação "NÃO" aplicada a uma operação "E" é igual à operação "OU" aplicada aos complementos das variáveis de entrada", ou seja

$$\overline{A.B} = \overline{A} + \overline{B}$$

2º) "A operação "NÃO" aplicada a uma operação "OU" é igual à operação "E" aplicada aos complementos das variáveis de entrada", ou seja

$$\overline{A + B} = \overline{A}.\overline{B}$$

Procure guardar as propriedades apresentadas acima, pois seu conhecimento é indispensável para o entendimento dos circuitos digitais.

2.4.7 - Porta "NÃO-E" como Bloco Lógico Universal

As operações "NÃO", "E" e "OU" podem ser realizadas utilizando-se portas "NÃO-E" e, por este motivo, a porta "NÃO-E" é conhecida como *bloco lógico universal* Se as três operações básicas podem ser realizadas com portas "NÃO-E", isto significa, em última análise, que *qualquer* circuito lógico pode ser sintetizado apenas com este tipo de porta. A importância prática deste fato decorre de serem as portas "NÃO-E" facilmente conseguidas na forma de circuitos integrados, com características de baixo custo e bom desempenho. Então, muitas vezes é mais

conveniente construir um circuito lógico utilizando apenas portas "NÃO-E" do que utilizar diversos tipos de portas lógicas.

Uma porta "NÃO-E" com duas entradas fornece a saída

$$Y = \overline{A.B}$$

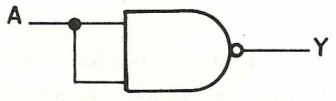
Aplicando o mesmo sinal às duas entradas (entrada B = entrada A), temos

$$Y = \overline{A.A}$$

mas como A.A = A, segue-se que

$$Y = \overline{A}$$

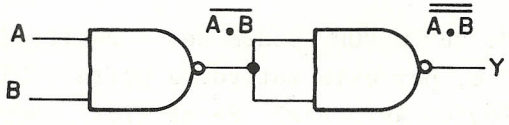
Em outras palavras, para obter a função "NÃO" a partir da porta "NÃO-E", basta ligar todas as entradas juntas. A figura 2.21 mostra um inversor construído a partir de uma porta "NÃO-E" de duas entradas.



$Y = \overline{A}$

Fig. 2.21 - Inversor a partir da porta "NÃO-E"

Como a operação "NÃO-E" é o inverso da operação "E", basta inverter a saída da porta "NÃO-E" para obter a função "E" (lembre-se do teorema da involução, ou dupla negação). A figura 2.22 mostra como é obtida a função "E" para duas variáveis a partir de portas "NÃO-E".



$Y = \overline{\overline{A.B}} = A.B$

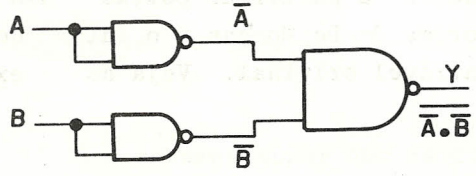
Fig. 2.22 - Função "E" a partir de portas "NÃO-E".

Para conseguir a função "OU" a partir de portas "NÃO-E", devemos considerar o primeiro teorema de De Morgan, lembrando também que a dupla negação retorna à variável original.

Veja as expressões abaixo:

$$Y = \overline{\bar{A}.\bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A + B$$

ou seja, para obter a função "OU", realizamos a operação "NÃO-E" com os complementos das variáveis de entrada. A figura 2.23 mostra como é obtida a função "OU" para duas variáveis a partir de portas "NÃO-E".



$$Y = \overline{\bar{A}.\bar{B}} = A + B$$

Fig. 2.23 - Função "OU" a partir de portas "NÃO-E".

2.4.8 - Porta "NÃO-OU" como Bloco Lógico Universal

Também a porta "NÃO-OU" pode ser utilizada para realizar as três operações lógicas fundamentais "NÃO", "E" e "OU". Assim, a porta "NÃO-OU" é também um *bloco lógico universal* e *qualquer* circuito lógico pode ser construído apenas com este tipo de porta. A importância deste fato decorre de serem as portas "NÃO-OU" também facilmente conseguidas na forma de circuitos integrados, com características de baixo custo e bom desempenho, tornando conveniente a construção de circuitos lógicos apenas com as portas "NÃO-OU".

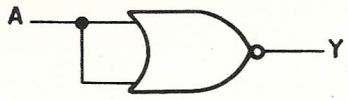
Se aplicarmos o mesmo sinal às entradas de uma porta "NÃO-OU" de, por exemplo, duas entradas, teremos na saída

$$Y = \overline{A + A}$$

mas, como $A + A = A$, segue-se que

$$Y = \bar{A}$$

Assim para conseguir a função "NÃO" a partir da porta "NÃO-OU", basta conectar juntas todas as entradas da porta. A figura 2.24 mostra um inversor construído com uma porta "NÃO-OU" de duas entradas.



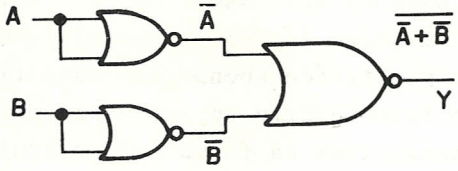
$$Y = \bar{A}$$

Fig. 2.24 - Inversor a partir da porta "NÃO-OU".

Para conseguir a função "E" a partir de portas "NÃO--OU", consideramos o segundo teorema de De Morgan e o fato de que a dupla negação retorna à variável original. Veja as expressões abaixo:

$$\begin{aligned} Y &= \overline{\bar{A} + \bar{B}} \\ &= \bar{\bar{A}} \,.\, \bar{\bar{B}} \\ &= A \,.\, B \end{aligned}$$

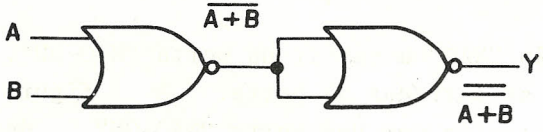
ou seja, para obter a função"E", realizamos a operação "NÃO-OU" com os complementos duas variáveis de entrada. A figura 2.25 mostra como é conseguida a função "E" para duas variáveis com portas "NÃO-OU".



$$Y = \overline{\bar{A} + \bar{B}} = A \,.\, B$$

Fig. 2.25 - Função "E" a partir de portas "NÃO-OU".

A obtenção da função "OU" a partir da função "NÃO-OU" é imediata,bastando inverter a saída da porta. A figura 2.26 ilustra como se consegue a função "OU" a partir de uma porta "NÃO-OU", para duas entradas.



$$Y = \overline{\overline{A + B}} = A + B$$

Fig. 2.26 - Função "OU" a partir de portas "NÃO-OU".

2.4.9 - Função "OU Exclusivo"

Uma importante função lógica de duas variáveis é a função "OU Exclusivo" (em inglês: EXCLUSIVE-OR ou EXOR), que encontra grande aplicação prática em circuitos digitais, como por exemplo somadores e geradores de paridade. A figura 2.27 mostra a tabela da verdade para a função "OU Exclusivo"; a operação "OU Exclusivo" é representada por uma cruz dentro de um círculo, conforme ilustra a figura 2.27.

A	B	Y
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	Ø

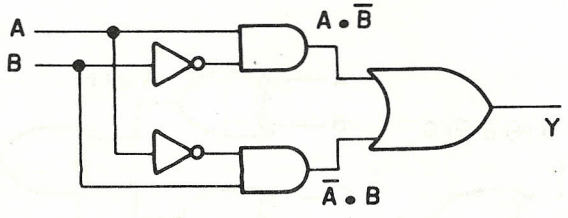
$Y = A \oplus B$

Fig. 2.27 - Tabela da Verdade para a função "OU Exclusivo".

Vemos, pela tabela, que a saída Y será igual a "1" se, e somente se, as variáveis de entrada forem diferentes. A função "OU Exclusivo" é expressa em termos das operações fundamentais pela equação:

$$Y = A.\bar{B} + \bar{A}.B$$

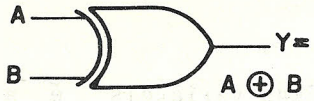
de modo que um circuito que realiza esta operação pode ser representado como mostrado na figura 2.28.

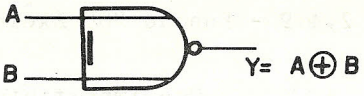


$Y = A.\bar{B} + \bar{A}.B$
$= A \oplus B$

Fig. 2.28 - Circuito "OU Exclusivo".

A importância do circuito "OU Exclusivo" é tal que existem símbolos especiais para representá-lo. A figura 2.29 ilustra os dois símbolos mais comuns.





SIMBOLOGIA AMERICANA SIMBOLOGIA EUROPÉIA

Fig. 2.29 - Símbolos de uma porta "OU Exclusivo".

Quando se deseja realizar a função "OU Exclusivo" para mais de duas variáveis, os blocos lógicos são interligados conforme ilustra a figura 2.30, que mostra a função "OU Exclusivo" para três variáveis. Fica como exercício para você a construção da tabela da verdade para este circuito.

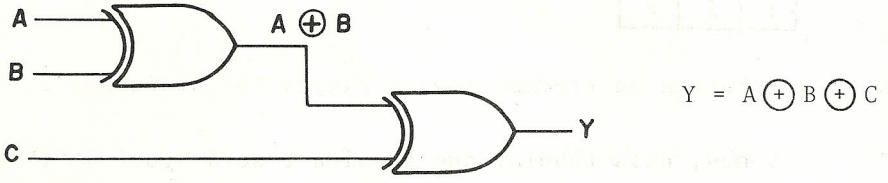


Fig. 2.30 - Interligação dos Blocos "OU-EXCLUSIVO" para três variáveis.

Para quatro variáveis temos as duas formas de interligação mostradas na figura 2.31. A forma (b) é preferível, porque envolve menos atrasos (apenas dois níveis, contra três da forma (a)).

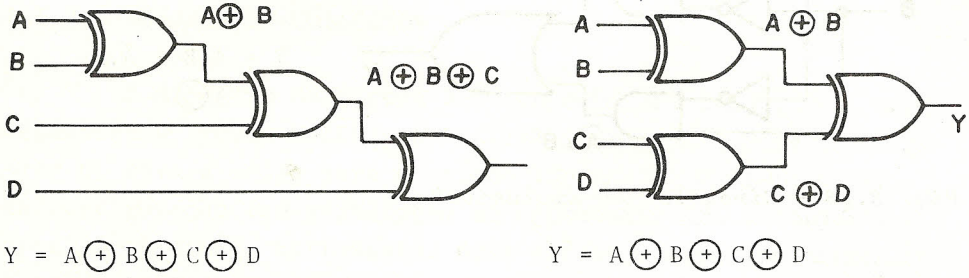


Fig. 2.31 - Interligação de Blocos "OU Exclusivo" para quatro variáveis.

2.4.10 - Função Coincidência

Outra importante função lógica de duas variáveis é a chamada "função Coincidência" ou "Equivalência" (em inglês: Coincidence, Equivalence, ou ainda Exclusive-NOR), cuja tabela da verdade é mostrada na figura 2.32. Esta função é representada por um ponto dentro de um círculo, conforme indicado na figura.

A	B	Y
Ø	Ø	1
Ø	1	Ø
1	Ø	Ø
1	1	1

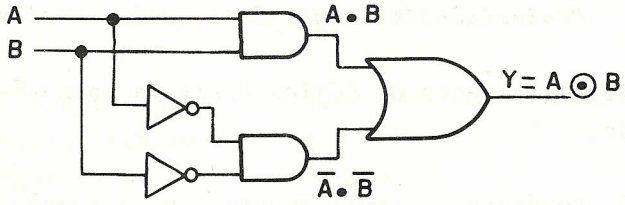
$Y = A \odot B$

Fig. 2.32 - Tabela da Verdade para a função "Coincidência".

Esta função é o inverso da função "OU Exclusivo" e sua saída Y será igual a "1" se, e somente se, as variáveis de entrada forem iguais. A função "Coincidência" é expressa em termos das operações fundamentais pela equação:

$$Y = A \cdot B + \overline{A} \cdot \overline{B}$$

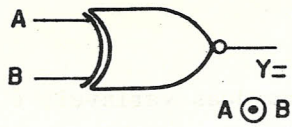
de modo que um circuito que realiza esta operação pode ser representado conforme mostra a figura 2.33.



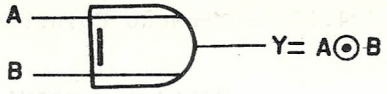
$$Y = A \cdot B + \overline{A} \cdot \overline{B} = A \odot B$$

Fig. 2.33 - Circuito "Equivalência" ou "Coincidência".

Existe também um símbolo especial para representar o circuito "Coincidência" ou "Equivalência". Os símbolos mais comuns estão mostrados na figura 2.34.



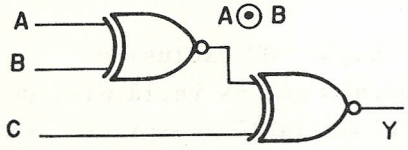
SIMBOLOGIA AMERICANA



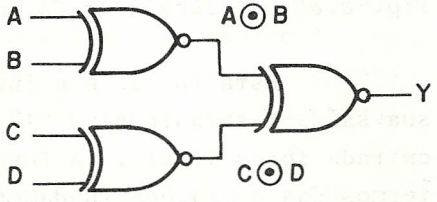
SIMBOLOGIA EUROPÉIA

Fig. 2.34 - Símbolos do Circuito "Coincidência".

Quando se deseja realizar a função "Coincidência" para mais de duas variáveis, os blocos lógicos são interligados conforme ilustra a figura 2.35, que mostra a interligação dos blocos lógicos para três e quatro variáveis. O levantamento das tabelas da verdade fica a seu cargo, a título de exercício.



$Y = A \odot B \odot C$



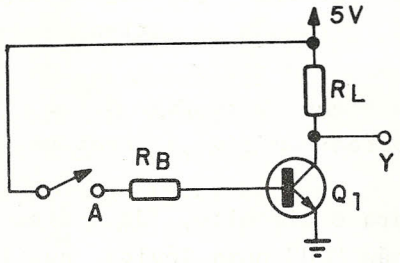
$Y = A \odot B \odot C \odot D$

Fig. 2.35 - Interligação dos Blocos Lógicos para a função "Coincidência" com três e quatro variáveis.

2.4.11 - Transformações de Lógica Positiva para Lógica Negativa.

Já comentamos, anteriormente, que a função lógica realizada por um circuito depende de utilizarmos lógica positiva ou lógica negativa. Nos diversos circuitos lógicos que apresentamos como exemplo, mostramos a tabela de estados para o circuito e a correspondente tabela da verdade que resulta ao utilizarmos lógica positiva. Você já deve ter percebido que a tabela da verdade resultante de estados seria outra caso utilizássemos lógica negativa. Vamos analisar o que aconteceria neste caso.

A figura 2.36 mostra um circuito lógico utilizando um transistor. Neste circuito, vemos que a tensão de saída Y será aproximadamente 5 Volts com a entrada A em Ø Volts, porque Q_1 está cortado, e aproximadamente Ø Volts com a entrada em 5 Volts, porque Q_1 estará saturado. A tabela de estados mostrada na figura reflete estes resultados.



A	Y
ØV	~5V
5V	~ØV

Fig. 2.36 - Circuito Lógico com Transistor.

As tabelas da verdade para lógica positiva e negativa referentes ao circuito da figura 2.36 estão mostradas na figura 2.37. Vemos que, independente do tipo de lógica utilizado, o circuito da figura 2.36 é um inversor lógico.

A	Y
Ø	1
1	Ø

(a) Lógica Positiva

A	Y
1	Ø
Ø	1

(b) Lógica Negativa

Fig. 2.37 - Tabelas da verdade para o Circuito da Fig. 2.36 para Lógica Positiva e Lógica Negativa.

Assim, concluímos que o circuito realiza a operação "NÃO" independentemente do tipo de lógica utilizado.

Considere agora o circuito da figura 2.7, cuja tabela de estados e as correspondentes tabelas da verdade para lógica positiva e negativa está ilustradas na figura 2.38. Então, este circuito executa a operação "E" para lógica positiva e a operação "OU" para lógica negativa.

A	B	Y
ØV	ØV	ØV
ØV	5V	ØV
5V	ØV	ØV
5V	5V	5V

(a) Tabela de Estados

A	B	Y
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

(b) Lógica Positiva

A	B	Y
1	1	1
1	Ø	1
Ø	1	1
Ø	Ø	Ø

(c) Lógica Negativa

Fig. 2.38 - Tabela de Estados para o circuito da fig. 2.7 e Tabelas da verdade para Lógica Positiva e Negativa.

Fazendo a mesma análise para o circuito da figura 2.11, vemos que ele realiza a operação "OU" para lógica positiva e a operação "E" para lógica negativa. A tabela de estados e as tabelas da verdade para os dois tipos de lógica estão ilustrados na figura 2.39.

A	B	Y
ØV	ØV	ØV
ØV	5V	5V
5V	ØV	5V
5V	5V	5V

(a) Tabela de Estados

A	B	Y
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

(b) Lógica Positiva

A	B	Y
1	1	1
1	Ø	Ø
Ø	1	Ø
Ø	Ø	Ø

(c) Lógica Negativa

Fig. 2.39 - Tabela de Estados para o circuito da fig. 2.11 e Tabelas da Verdade para Lógica Positiva e Negativa.

As tabelas de estados e as tabelas da verdade com lógica positiva e negativa para os circuitos das figuras 2.15 e 2.19 estão ilustradas na figura 2.39. Vemos que o primeiro realiza a função "NÃO-E" para lógica positiva e "NÃO-OU" para lógica negativa, ao passo que o segundo realiza a operação "NÃO-OU" para lógica positiva e "NÃO-E" para lógica negativa.

A	B	Y
ØV	ØV	∿5V
ØV	5V	∿5V
5V	ØV	∿5V
5V	5V	∿ØV

(a) Tabela de Estados

A	B	Y
Ø	Ø	1
Ø	1	1
1	Ø	1
1	1	Ø

(b) Lógica Positiva

A	B	Y
1	1	Ø
1	Ø	Ø
Ø	1	Ø
Ø	Ø	1

(c) Lógica Negativa

A	B	Y
ØV	ØV	∿5V
ØV	5V	∿ØV
5V	ØV	∿ØV
5V	5V	∿ØV

(a) Tabela de Estados

A	B	Y
Ø	Ø	1
Ø	1	Ø
1	Ø	Ø
1	1	Ø

(b) Lógica Positiva

A	B	Y
1	1	Ø
1	Ø	1
Ø	1	1
Ø	Ø	1

(c) Lógica Negativa

Fig. 2.40 - Tabelas de Estado e Tabelas da Verdade com Lógica Positiva e Negativa para os Circuitos das Figs. 2.15 e 2.19.

Todos estes resultados estão resumidos sob forma de tabela na figura 2.41, que mostra as transformações de lógica positiva para lógica negativa.

LÓGICA POSITIVA	LÓGICA NEGATIVA
NÃO	NÃO
E	OU
OU	E
NÃO-E	NÃO-OU
NÃO-OU	NÃO-E

Fig. 2.41 - Transformações de Lógica Positiva para Negativa.

CAPÍTULO 3 - FAMÍLIAS LÓGICAS

3.1 - Introdução

Estudamos até o momento as diversas operações lógicas sem nos preocuparmos muito com os elementos utilizados para implementá-las na prática. De fato, a menos dos circuitos apresentados a título de exemplo, não discutimos com maior profundidade de que maneira e utilizando quais dispositivos construimos os circuitos digitais.

Os assuntos discutidos no capítulo 2 são absolutamente gerais e as funções lógicas estudadas podem ser implementadas com elementos discretos (relés, diodos, transistores, etc) ou *circuitos integrados* (CI'S).

Circuitos integrados são dispositivos semicondutores nos quais uma única pastilha (em inglês: CHIP) de silício integra um certo número de resistores, diodos, transistores,etc, número este que pode ser elevadíssimo, ultrapassando a casa dos milhares.

De um modo geral, podemos afirmar que os circuitos integrados apresentam, em relação aos circuitos com elementos discretos, as seguintes vantagens:

a) menor tamanho,
b) menor consumo de potência,
c) menor custo,
d) maior velocidade,
e) maior confiabilidade.

Devido a todas estas vantagens, os circuitos digitais em sua quase totalidade são construídos com CI'S.

3.2 - Famílias de Circuitos Integrados

De acordo com a tecnologia de construção, os circuitos integrados são agrupados em famílias, entre as quais podemos citar:

a) RTL - Resistor Transistor Logic
b) RCTL - Resistor Capacitor Transistor Logic
c) DTL - Diode Transistor Logic
d) TTL - Transistor Transistor Logic
e) C-MOS - Complementary Metal Oxide Semiconductor
f) ECL - Emitter Coupled Logic

As três primeiras são as mais antigas e se encontram hoje em desuso. A família ECL é a mais veloz de todas, mas seu emprego se encontra restrito a aplicações que requeiram altíssimas velocidades de comutação, devido ao alto custo e às dificuldades que seu uso apresenta.

Restam, assim, as famílias TTL e C-MOS, que são as mais utilizadas atualmente.

Neste capítulo, faremos um estudo detalhado das características destas duas famílias e, em capítulos posteriores, apresentaremos grande número de aplicações empregando CI'S TTL e C-MOS.

3.3 - Família TTL

3.3.1 - Visão Geral

A grande maioria dos circuitos integrados TTL pertence às séries 54 e 74, introduzidas originalmente pela Texas Instruments e que são hoje um padrão da indústria, fornecidas por diversos fabricantes.

A série 54 é de uso militar e opera na faixa de temperatura de -55^0C a $+125^0C$, com uma tensão de alimentação de $5 \pm 0,5V$.

A série 74 é de uso geral, operando na faixa de temperatura de 0^0C a $+70^0C$, com alimentação de $5 \pm 0,25V$.

Há centenas de funções disponíveis nas séries 54/74, abrangendo portas, flip-flops, decodificadores, contadores, etc. Conforme o número de portas contidas em um, CI ele se classifica como:

- SSI (SMALL SCALE INTEGRATION, ou integração em pequena escala): de 1 a 12 portas lógicas;
- MSI (MEDIUM SCALE INTEGRATION, ou integração em média escala): de 13 a 99 portas lógicas;
- LSI (LARGE SCALE INTEGRATION, ou integração em grande escala): de 1ØØ a 1ØØØ portas lógicas;
- VLSI (VERY LARGE SCALE INTEGRATION, ou integração em escala muito grande): acima de 1ØØØ portas lógicas.

Além da série 54/74, que é a mas importante e que possui o maior número de funções disponíveis, existem algumas outras séries como a 4ØØØMTTL da Motorola e a 82ØØ da Signetics.

3.3.2 - Características

A figura 3.1 mostra o diagrama de uma porta NAND de duas entradas da série 74, na versão standard.

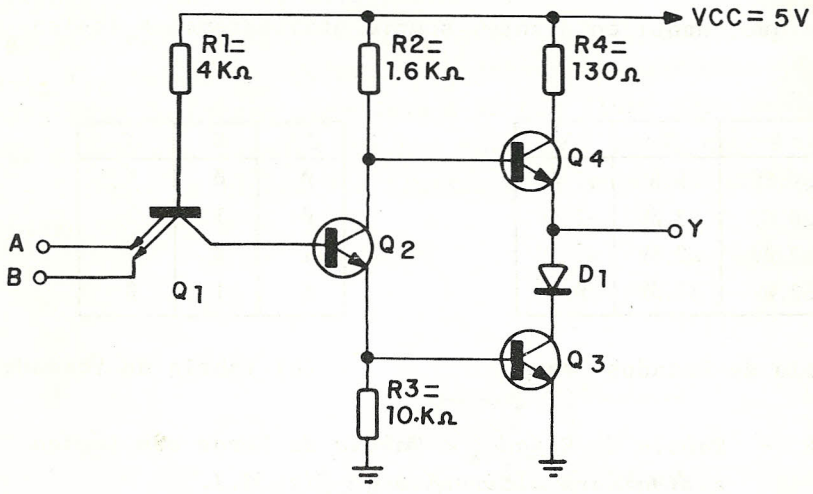


Fig. 3.1 - Porta NAND TTL de duas entradas.

Este é o elemento básico dos circuitos integrados

TTL. Observe que Q_1 é um transistor especial, com vários emissores, chamado multiemissor.

Suponhamos que as entradas A e B estejam no nível lógico "1", que corresponde a uma tensão $\geq 2,\emptyset$ V para este circuito. Nesta situação, Q_1 não se comporta como um transistor e sua junção base-coletor está diretamente polarizada. A corrente que flui através desta junção e para a base de Q_2 é suficiente para saturá-lo. Com a saturação de Q_2, ocorre o corte de Q_4 e a saturação de Q_3, fazendo com que a tensão de saída Y fique próxima de $\emptyset$ V (na realidade $\leq \emptyset,4V$).

Se a entrada A ou a entrada B (ou ambas) forem colocadas no estado lógico "$\emptyset$", correspondente a uma tensão $\leq \emptyset,8V$, a corrente de base de Q_2 será desviada para a entrada e Q_2 irá para o corte. Isto fará com que Q_4 sature e Q_3 corte, levando a tensão de saída Y para um valor $\geq 2,4V$, tipicamente igual a 3,3V.

Fazendo a tabela de estados e a tabela da verdade para lógica positiva, conforme ilustra a figura 3.2, concluímos que, de fato, o circuito da figura 3.1 é uma porta NAND para lógica positiva (POSITIVE NAND GATE). Todos os circuitos integrados TTL são descritos em termos de lógica positiva e fica estabelecido que, daqui em diante, sempre utilizaremos a lógica positiva.

A	B	Y
$\leq \emptyset,8V$	$\leq 0,8V$	$\geq 2,4V$
$\leq \emptyset,8V$	$\geq 2,\emptyset V$	$\geq 2,4V$
$\geq 2,\emptyset V$	$\leq \emptyset,8V$	$\geq 2,4V$
$\geq 2,\emptyset V$	$\geq 2,\emptyset V$	$\leq \emptyset,4V$

(a) Tabela de Estados

A	B	Y
$\emptyset$	$\emptyset$	1
$\emptyset$	1	1
1	$\emptyset$	1
1	1	$\emptyset$

(b) Tabela da Verdade

Fig. 3.2 - Tabela de Estados e Tabela da Verde com Lógica Positiva para o Circuito da fig. 3.1.

Observe que a situação descrita quando as entradas estão no nível lógico "1" se repetiria se as mesmas estivessem

abertas. Isto significa que *um CI TTL interpreta uma entrada aberta como estado lógico "1"* ;entretanto, *nunca* se deve deixar uma entrada aberta porque isto pode acarretar problemas de ruído.

Vamos definir alguns parâmetros que são importantes para os CI'S da família TTL.

a) V_{IL} - é a tensão de entrada que o circuito interpreta como nível lógico "Ø" ; para o circuito da fig. 3.1 $V_{IL} \leq Ø,8V$.

b) V_{IH} - é a tensão de entrada que o circuito interpreta como nível lógico "1"; para o circuito da figura 3.1 $V_{IH} \geq 2,ØV$

c) V_{OL} - é a tensão de saída no estado lógico "Ø"; na figura 3.1 $V_{OL} \leq Ø,4V$

d) V_{OH} - é a tensão de saída no estado lógico "1"; na figura 3.1 $V_{OH} \geq 2,4V$, sendo 3,3V um valor típico

e) I_{IL} - Quando uma entrada está no nível lógico "Ø" flui uma corrente da base para o emissor de Q_1 (sentido convencional)e o dispositivo que está forçando o nível "Ø" tem que ser capaz de absorver esta corrente, denominada I_{IL}. No circuito da figura 3.1, $I_{IL} \leq 1,6$ mA. Convenciona-se que as correntes que fluem *para fora* do CI são *negativas* e as que fluem *para dentro* dele são *positivas*, de modo que nos manuais é comum encontrar-se $I_{IL} \leq -1,6$ mA. A figura 3.3 ilustra a situação de uma entrada no estado lógico "Ø".

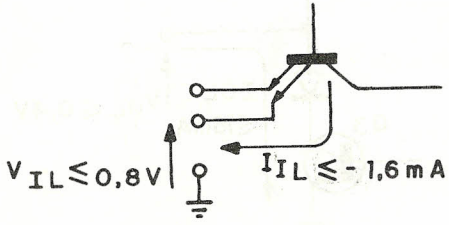


Fig. 3.3 - Entrada no Estado Lógico "Ø".

f) I_{IH} - Quando uma entrada está no nível lógico "1" flui uma pequena corrente do emissor para a base de Q_1 e o dispositivo que força o nível "1" nesta entrada tem que ser capaz de fornecer esta corrente, denominada I_{IH}. Para a figura 3.1, $I_{IH} \leq 40\mu A$. A figura 3.4 mostra a situação de uma entrada no estado lógico "1".

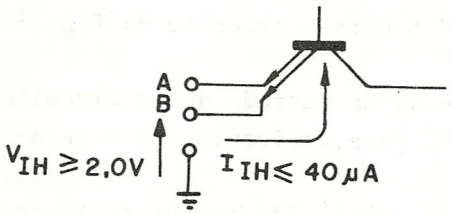


Fig. 3.4 - Entrada no Estado Lógico "1".

g) I_{OL} - Quando a saída do circuito da figura 3.1 vai para o estado lógico "Ø", o transistor Q_3 está saturado. O circuito que estiver ligado a esta saída fornecerá então uma corrente que tem que ser absorvida por Q_3, denominada I_{OL}. O valor máximo de I_{OL}, sem que ocorra a degradação do nível lógico "Ø", é de 16 mA. A fig. 3.5 mostra a situação da saída no estado lógico "Ø".

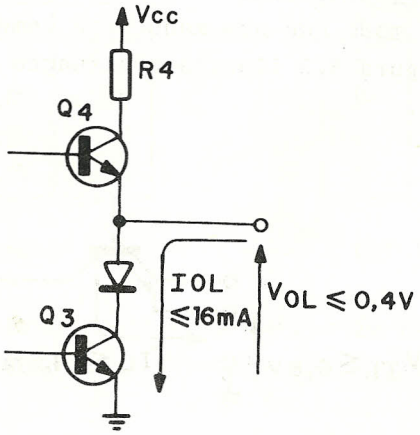


Fig. 3.5 - Saída no Estado Lógico "1".

h) I_{OH} - Quando a saída está no estado lógico "1", Q_3 está cortado e Q_4 saturado. Circula então através de Q_4 uma corrente que é fornecida ao circuito ligado à saída. Esta corrente I_{OH} deve ser no máximo igual a 4ØØ µA (ou - 4ØØ µA, considerando o sentido) para que não ocorra a degradação do nível lógico "1". A figura 3.6 ilustra a situação da saída no estado "1".

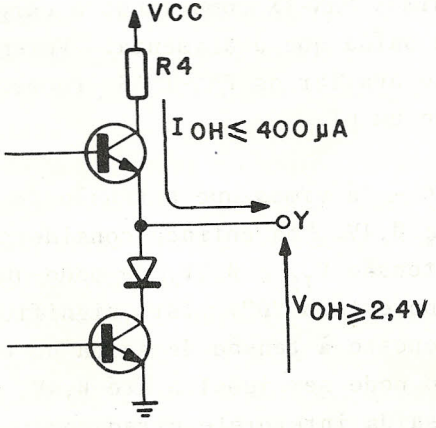


Fig. 3.6 - Saída no Estado Lógico "1".

i) FAN OUT - Numa aplicação prática, a saída de uma porta estará ligada às entradas de diversas outras. Conforme vimos anteriormente, há uma série de correntes envolvidas (I_{IL}, I_{IH}, I_{OL} e I_{OH}) e devemos examinar as relações entre elas.

No estado lógico "Ø", a saída de uma porta é capaz de drenar no máximo 16 mA do circuito a ela ligado, ao passo que uma entrada no estado lógico "Ø" fornece uma corrente de no máximo 1,6 mA. Concluímos que uma saída é capaz de absorver a corrente fornecida por até 1Ø entradas e dizemos que o FAN-OUT do circuito da figura 3.1 é igual a 1Ø no estado lógico "Ø".

No estado lógico "1", uma saída fornece no máximo 4ØØ µA, enquanto uma entrada drena no máximo 4Ø µA. Novamente, a saída é capaz de fornecer corrente para até 1Ø entradas, e assim dizemos que o FAN-OUT é igual a 1Ø no estado lógico "1".

De forma resumida podemos definir o FAN-OUT como o número máximo de entradas que pode ser ligado a uma saída; observe que o FAN-OUT pode ser diferente nos estados "Ø" e "1", como

veremos em outros itens. É importante observar que só tem sentido falar em FAN-OUT quando se trata da interligação de CI'S da mesma família. Cumpre, ainda, notar que as tensões V_{OH} e V_{OL} são garantidas com os valores $V_{OH} \geq 2,4V$ e $V_{OL} \leq Ø,4V$ para a saída com carga máxima, ou seja, a condição mais severa (WORST--CASE CONDITION).

j) FAN-IN - Definimos FAN-IN como sendo a carga que uma entrada representa para a saída que a alimenta. Veremos mais adiante um modo prático de avaliar os FAN-IN'S representados pelas diversas entradas de um CI.

k) Margem de Ruído - Já vimos que a tensão de saída no estado lógico "Ø" é $V_{OL} \leq Ø,4V$. Uma entrada considera como estado lógico "Ø" qualquer tensão $V_{IL} \leq Ø,8V$, de modo que há uma margem de ruído de Ø,4V no estado "Ø". Isto significa que, se tivermos um ruído sobreposto à tensão de saída de uma porta no estado "Ø", este ruído pode ser igual a até Ø,4V, sem que uma entrada ligada a esta saída interprete erradamente o sinal lógico. Do mesmo modo, com a saída no estado "1", temos $V_{OH} \geq 2,4V$, ao passo que uma entrada interpreta como estado "1" qualquer tensão $V_{IH} \geq 2,ØV$. Novamente, a margem de ruído no estado "1" é igual a Ø,4V.

A figura 3.7 ilustra as margens de ruído para os circuitos integrados TTL.

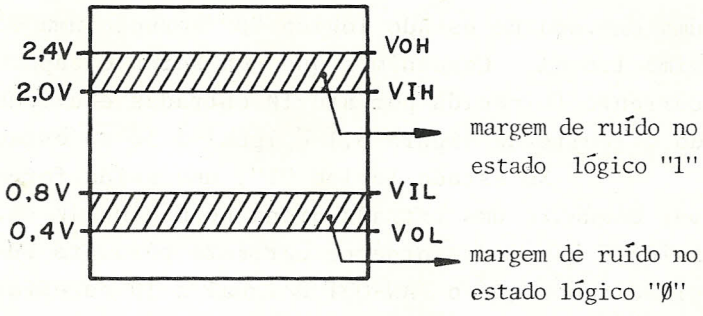


Fig. 3.7 - Margens de Ruído.

1) Característica de Transferência - Uma pergunta que surge naturalmente neste ponto é o que ocorre quando a tensão de entrada V_I tem um valor maior que Ø,8V (V_{IL}) e menor que 2,ØV (V_{IH}). Esta condição é transitória e só se apresenta quando a entrada está passando de um estado lógico para outro. Imagine que no circuito da fig. 3.1 as entradas A e B estejam ligadas juntas e que a tensão de entrada V_I seja variada entre "Ø" e 5V. A tensão de saída V_0 irá variar de acordo com a curva de transferência ilustrada na figura 3.8.

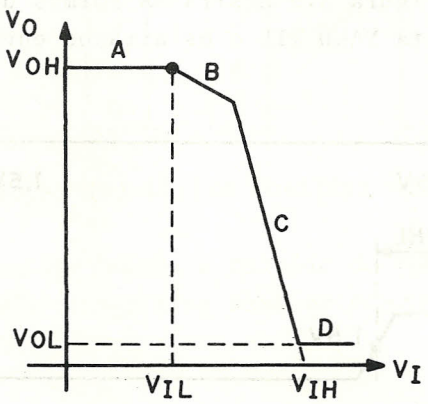


Fig. 3.8 - Curva de Transferência.

Nesta curva distinguimos 4 regiões distintas

- região A: enquanto a tensão de entrada está menor que ≈ Ø,7V, Q_2 e Q_3 estão cortados e Q_4 está saturado;
- região B: aumentando a tensão de entrada, Q_2 entra na região ativa e a tensão de saída decresce porque aumenta a corrente através de R_4;
- região C: aumentando ainda mais a tensão de entrada, Q_3 também conduz e a tensão de saída decresce rapidamente;
- região D: Q_4 corta e a tensão de saída fica independente da de entrada.

No funcionamento normal apenas as regiões A e D são permanentes; as regiões B e C correspondem à comutação da porta.

Observe que durante a comutação há um instante em que os transistores Q_2, Q_3 e Q_4 estão conduzindo ao mesmo tempo e a corrente drenada da fonte de alimentação aumenta bruscamente. Isto gera picos na corrente de alimentação Icc, que exigem cuidados especiais para evitar problemas. Discutiremos este assunto em um item posterior.

m) Atraso de Propagação - Já observamos no capítulo 2 que a tensão de saída de uma porta nunca responde instantaneamente às variações de entrada. Há sempre um certo atraso associado à porta lógica. A figura 3.9 mostra as formas de onda de entrada e saída de uma porta NAND TTL e os atrasos envolvidos.

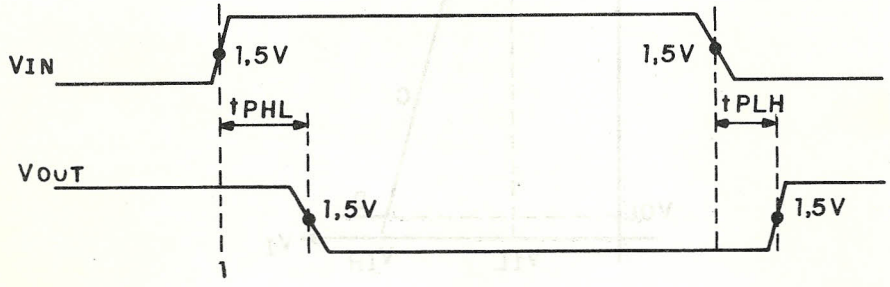


Fig. 3.9 - Formas de Onda de Entrada e Saída e Atrasos.

t_{pHL} é o tempo de transição do estado lógico "1" para o estado lógico "Ø", medido da forma indicada. t_{pHL} é tipicamente igual a 7 ns.

t_{pHL} é o tempo de transição do estado lógico "Ø" para o estado lógico "1", medido da forma indicada. t_{pHL} é tipicamente igual a 11 ns.

Define-se o atraso de propagação t_{pd} como a média aritmética de t_{pHL} e t_{pLH}.

$$t_{pd} = \frac{t_{pHL} + t_{pLH}}{2}$$

resultando num valor típico de 9 ns.

O atraso de propagação é um fator que limita a aplicação de um CI, porque se a entrada varia de modo excessivamente rápido, a saída simplesmente não consegue acompanhar as variações da entrada e o funcionamento torna-se errático; a família TTL é uma das mais velozes existentes.

n) Dissipação de Potência - A corrente drenada da fonte pela porta lógica dependerá de estar a saída em "Ø" ou em "1". I_{CCL} é a corrente consumida com a saída no estado lógico "Ø" e I_{CCH} a corrente consumida com a saída no estado "1". Para a porta lógica da figura 3.1 I_{CCL} é de no máximo 5,5 mA (3 mA típico) e I_{CCH} é de no máximo 2 mA (1 mA típico).

A corrente média, assumindo que a saída fica 50% do tempo em cada estado, é Icc = 2 mA (típico), resultando numa dissipação de potência de 10 mW (5V x 2 mA).

3.3.3 - Tipos de Circuitos Integrados TTL

O circuito da figura 3.1 representa uma porta TTL na versão standard. Introduzindo algumas modificações neste circuito, foram desenvolvidas outras versões, todas compatíveis entre si, de modo a obter características melhores em um ou outro ponto.

Assim, dentro da série 54/74 encontramos cinco versões distintas:

- 54/74 - Standard,
- 54L/74L - LOW POWER (baixa potência),
- 54H/74H - HIGH SPEED (alta velocidade)
- 54S/74S - SCHOTTKY,
- 54LS/74LS - LOW POWER SCHOTTKY.

A figura 3.10 mostra um quadro comparativo com as características principais das 5 versões:

PARÂMETRO	74L	74LS	74	74H	74S
Atraso de Propagação, tpd (ns)	33	10	10	6	3
Dissipação de Potência/GATE (mW)	1	2	10	22	20

Fig. 3.10 - Versões de TTL.

A versão standard é a de mais baixo custo e a que possui a maior variedade de funções disponíveis. É importante ressalvar que nem todos os CI'S estão disponíveis em todas as versões.

A versão LOW POWER, indicada pela letra L (por exemplo: 74LØØ) no nome do CI apresenta o mais baixo consumo de potência, às expensas de uma redução na velocidade. Esta versão é indicada nas aplicações onde o baixo consumo seja o fator mais importante, como em equipamentos operados com baterias, e a velocidade requerida não seja muito alta.

A versão HIGH SPEED, indicada pela letra H no nome do CI (por exemplo 74HØØ) apresenta uma velocidade maior que a standard, com um consumo bem mais elevado. Esta versão encontra pouco uso atualmente, sendo substituída pela S.

A versão SCHOTTKY é a mais veloz de todas. Os circuitos integrados desta versão têm uma letra S em seu nome (por exemplo: 74SØØ); esta versão é indicada nas aplicações que requeiram altas velocidades de comutação.

A versão LOW POWER SCHOTTKY, indicada pelas letras LS no nome do CI (por exemplo; 74LSØØ) é a mais recente de todas e oferece a mesma velocidade da versão standard com um consumo bem menor. Sob este aspecto é a versão LS uma solução quase ótima na maioria dos casos.

A escolha de uma ou outra versão depende dos requisitos particulares de cada circuito. Devem-se levar em conta os fatores de custo, velocidade e consumo de potência, analisando cuidadosamente estes aspectos para se encontrar a solução mais adequada em cada caso.

3.3.4 - Interligação dos Diversos Tipos de TTL

Todas as cinco versões da série 54/74 são compatíveis entre si e operam a partir da mesma alimentação de 5V.

Entretanto, é preciso considerar as características de cada uma para determinar o número máximo de entradas de uma versão que podem ser ligadas à saída de outra. A figura 3.11 mostra as características de tensão e corrente para os diversos tipos de TTL:

	74L	74	74LS	74H	74S	UNIDADE
V_{OL}	Ø,3	Ø,4	Ø,5	Ø,4	Ø,5	V
V_{OH}	2,4	2,4	2,7	2,4	2,7	V
I_{OL}	4,Ø	16	8,Ø	2Ø	2Ø	mA
I_{OH}	2ØØ	4ØØ	4ØØ	5ØØ	1ØØØ	µA
V_{IL}	Ø,7	Ø,8	Ø,8	Ø,8	Ø,8	V
V_{IH}	2,Ø	2,Ø	2,Ø	2,Ø	2,Ø	V
I_{IL}	Ø,18	1,6	Ø,36	2,Ø	2,Ø	mA
I_{IH}	1Ø	4Ø	2Ø	5Ø	5Ø	µA

Fig. 3.11 - Características das Diversas Versões da Família TTL.

Através desta tabela é fácil determinar os FAN-OUT'S de cada tipo. Lembre-se de que o FAN-OUT só tem sentido para interligação de CI'S *do mesmo tipo*.

O FAN-OUT no estado lógico "Ø" será

$$I_{OL} \div I_{IL}$$

e, no estado alto,

$$I_{OH} \div I_{IH}$$

Fazendo estes cálculos obtemos a tabela mostrada na figura 3.12:

FAN-OUT	74L	74	74LS	74H	74S
Estado "Ø"	22	1Ø	22	1Ø	1Ø
Estado "1"	2Ø	1Ø	2Ø	1Ø	2Ø

Fig. 3.12 - FAN-OUT'S para os Diversos Tipos .

Quando os FAN-OUT'S nos estados lógicos "1" e "Ø" são diferentes, devemos guiar-nos pelo menor valor, a não ser que saibamos de antemão que a saída se manterá sempre no estado correspondente ao maior valor. Por exemplo, a saída de uma porta S é capaz de drenar a corrente de 1Ø entradas S, quando no estado lógico "Ø" , ou fornecer corrente para 2Ø entradas S, quando no estado lógico "1". Então, devemos ligar no máximo 1Ø entradas a uma saída, a não ser que esta saída seja usada apenas para fixar um nível "1" nas entradas e não varie, caso em que poderemos ligar até 2Ø entradas.

Vamos supor que queiramos ligar a saída de uma porta LS a entradas S.
No estado lógico "Ø" temos:

$$\frac{I_{OL}\ (LS)}{I_{IL}\ (S)} = \frac{8,Ø}{2,Ø} = 4$$

no estado lógico "1":

$$\frac{I_{OH}\ (LS)}{I_{IH}\ (S)} = \frac{4ØØ}{5Ø} = 8$$

Tomamos o menor valor e dizemos que uma saída LS pode ser ligada a, no máximo, 4 entradas S.

Fazendo estes cálculos para todas as combinações possíveis obtemos a tabela mostrada na figura 3.13.

Nº MAXIMO DE ENTRADAS	SAÍDA					
		74L	74	74LS	74H	74S
	74L	2Ø	4Ø	4Ø	5Ø	1ØØ
	74	2,5	1Ø	5	12,5	12,5
	74LS	1Ø	2Ø	2Ø	25	5Ø
	74H	2	8	4	1Ø	1Ø
	74S	2	8	4	1Ø	1Ø

Fig. 3.13 - Número Máximo de Entradas Permissíveis para uma Saída.

Pela tabela, temos, por exemplo, que uma saída standard é capaz de alimentar 4Ø entradas L, ou 1Ø entradas standard, ou 2Ø entradas LS, ou 8 entradas H, ou 8 entradas S.

Quando ligamos entradas de diferentes versões a uma saída, devemos computar as correntes e verificar se a capacidade da saída não é ultrapassada. Um modo simples de fazer isto é considerar a carga que uma entrada representa para a saída que a alimenta. Por exemplo:

Uma saída LS pode alimentar 4Ø entradas L, logo cada entrada L representa $^1/_{40}$ = Ø,Ø25 unidades de carga para esta saída.

Fazendo o cálculo para todas as combinações possíveis e multiplicando os valores por 1ØØ construímos a tabela ilustrada na figura 3.14:

		SAÍDA				
		74L	74LS	74	74H	74S
ENTRADA	74L	5	2,5	2,5	2	1
	74LS	1Ø	5,Ø	5	4	2
	74	4Ø	2Ø	1Ø	8	8
	74H	5Ø	25	12,5	1Ø	1Ø
	74S	5Ø	25	12,5	1Ø	1Ø

Fig. 3.14 - Tabela de Carga Normalizada.

Agora, para verificarmos se a saída é capaz de fornecer corrente para todas as entradas a ela conectadas, basta somarmos as cargas normalizadas. O valor total não pode ultrapassar 1ØØ.

A figura 3.15 ilustra exemplo deste método:

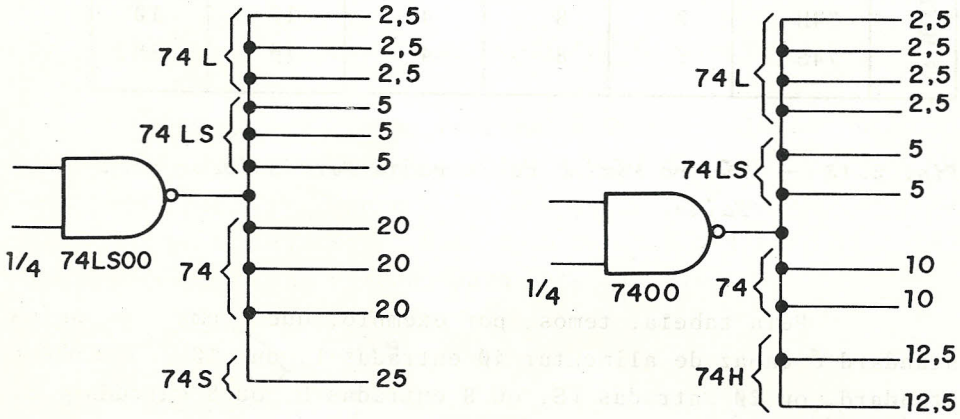


Fig. 3.15 - Exemplos de Interligação entre Diversas versões e Cálculo do Carregamento.

3.3.5 - Configurações de Saída

3.3.5.1 - Saída OPEN-COLLECTOR

O circuito da figura 3.1 tem uma configuração de saída denominada "TOTEM POLEM" ou "ACTIVE PULL-UP", porque o transistor Q_4 leva a saída para o estado lógico "1" (PULL-UP ACTION). Esta configuração tem a vantagem de apresentar uma velocidade de comutação elevada. Entretanto, duas saídas deste tipo nunca devem ser ligadas ao mesmo ponto. A figura 3.16 mostra uma ligação não permitida para saídas do tipo TOTEM-POLE:

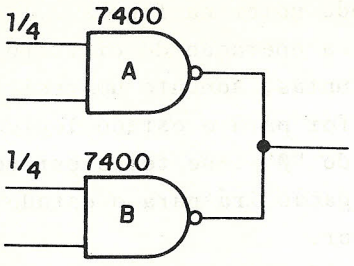


Fig. 3.16 - Ligação Não-Permitida para Saídas TOTEM-POLE.

Vamos supor que as entradas da porta A resultassem numa saída em "1", enquanto as entradas da porta B resultassem numa saída igual a "Ø". Na porta A, Q_4 estaria saturado enquanto que na porta B, Q_3 estaria saturado. Q_3 é capaz de absorver um máximo de 16 mA ($I_{OL} \leq 16$ mA), ao passo que Q_4 pode fornecer até 5Ø mA (I_{OS} - corrente com a saída em curto, dado de manual). Então, Q_3 da porta B seria provavelmente danificado nesta situação. As saídas do tipo TOTEM-POLE somente podem ser ligadas juntas quando se puder garantir que elas nunca tenderão a estados lógicos distintos.

Há um tipo de portas que permite a ligação de saídas ao mesmo ponto. A configuração de saída, neste caso, chama-se OPEN COLLECTOR (coletor aberto). A figura 3.17 ilustra o diagrama de uma porta NAND com a saída OPEN-COLLECTOR:

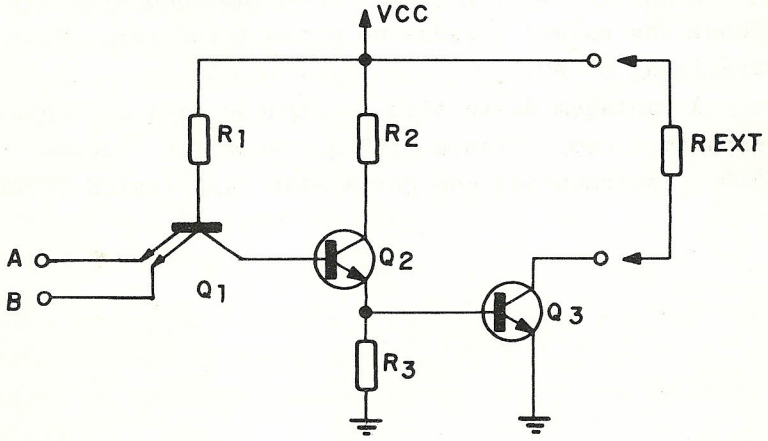


Fig. 3.17 - Porta NAND com a Saída OPEN-COLLECTOR.

Como se pode notar na figura 3.17, um resistor externo é necessário para a operação do circuito. Quando diversas saídas são ligadas juntas, somente um resistor é necessário. Se qualquer das saídas for para o estado lógico "Ø" o ponto de ligação ficará no estado "Ø" ; se todas estiverem no estado lógico "1" o ponto de ligação irá para o estado "1" devido à ação de PULL-UP do resistor.

A figura 3.18 mostra a ligação de portas NAND OPEN-COLLECTOR (7401) e a função lógica resultante:

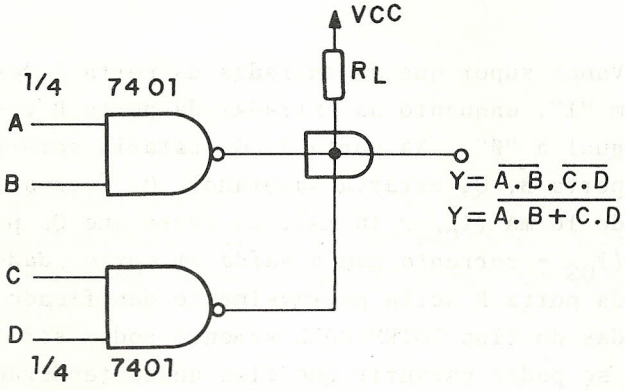


Fig. 3.18 - Ligação de Portas NAND OPEN-COLLECTOR e Função Resultante.

Este tipo de ligação é conhecido como WIRE-AND, pois se qualquer das saídas ligadas no ponto Y for para "Ø" o resultado será igual a "Ø".

A vantagem deste tipo de ligação fica evidente quando a comparamos com a figura 3.19 que mostra a mesma função $Y=\overline{A.B.C.D}$ implementada com porta NAND 74ØØ (saída TOTEM-POLE):

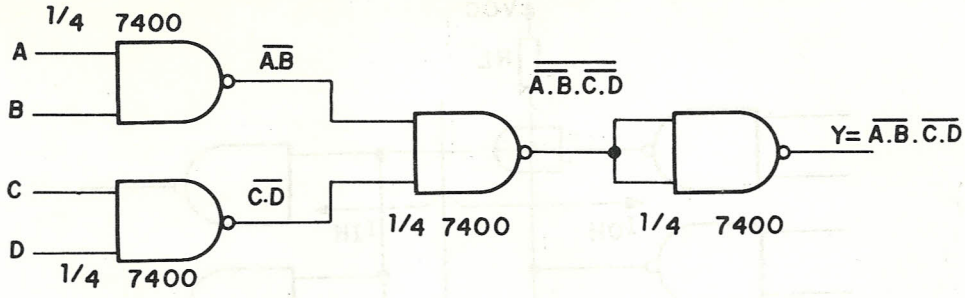


Fig. 3.19 - Função $Y = \overline{A.B}.\overline{C.D}$ *construída com portas 7400 (saída da TOTEM-POLE).*

Vemos que a ligação WIRE-AND permite economizar dois níveis de lógica, o que pode representar uma redução de custo acentuada em circuitos mais complexos.

A desvantagem da saída OPEN-COLLECTOR é a menor velocidade de comutação. Enquanto na configuração TOTEM POLE, o transistor Q_4 apresenta uma impedância muito pequena quando saturado, permitindo a transição rápida da saída do estado "Ø" para o estado "1", na configuração OPEN-COLLECTOR o resistor externo apresenta uma impedância significativamente maior, reduzindo a velocidade de comutação. Compare os valores típicos de t_{pHL} e t_{pLH} para duas portas NAND com configurações de saída TOTEM-POLE e OPEN-COLLECTOR.

7400 (TOTEM-POLE): $t_{pLH} = 11ns$ $t_{pHL} = 7ns$

7401 (OPEN-COLLECTOR): $t_{pLH} = 35ns$ $t_{pHL} = 8ns$

Concluímos que as portas com saída OPEN-COLLECTOR não se prestam a aplicações que requeiram grandes velocidades de comutação.

Veremos agora como calcular o valor de R_L para a ligação WIRE-AND. A figura 3.20 ilustra as condições para o cálculo de R_L no estado lógico "1":

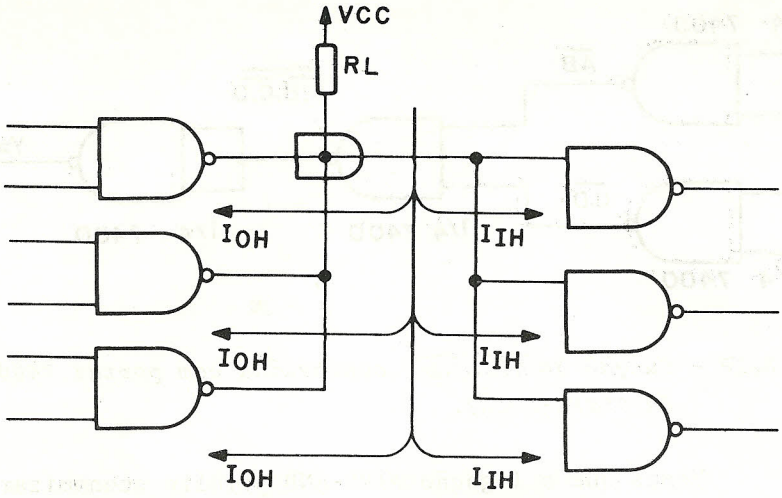


Fig. 3.20 - Cálculo de R_L no Estado Lógico "1".

O valor da tensão de saída desejado no estado lógico "1", V_{OH} (desejado) impõe um valor máximo para R_L. Se tivermos N saídas ligadas juntas e n entradas ligadas à conexão WIRE-AND, a corrente através de R_L será:

$$I_{RL} = N \cdot I_{OH} + n.I_{IH}$$

A queda de tensão em R_L será:

$$E_{RL} = V_{cc} - V_{OH} \text{ (desejado)},$$

logo:

$$R_{L(max)} = \frac{V_{cc} - V_{OH} \text{ (desejado)}}{N.I_{OH} + n.I_{IH}} \qquad (1)$$

Se utilizarmos um valor de R_L maior que o calculado pela expressão (1), haverá uma degradação do estado lógico "1", com a conseqüente diminuição da margem de ruído. No circuito

da figura 3.20, N=3 e n=3 . I_{OH} é 25Ø μA para portas OPEN-COLLECTOR e I_{IH}=4Ø μA. Então, para um V_{OH} (desejado) de 2,4V teríamos:

$$R_L \text{ (max)} = \frac{5 - 2,4}{3x\text{Ø},25 - 3x\text{Ø},\text{Ø}4} = 2,989 \text{ K}\Omega$$

A figura 3.21 mostra as condições para o cálculo de R_L no estado lógico "Ø". O estado "Ø" impõe uma limitação ao valor mínimo de R_L, que deve limitar a corrente ao valor I_{OL}, a menos que se possa garantir que mais de uma saída estará em "Ø" em todas as situações:

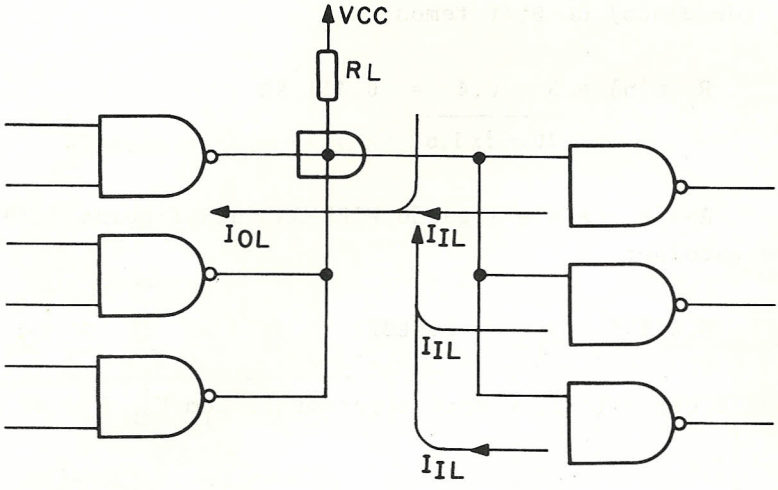


Fig. 3.21 - Cálculo de R_L no Estado Lógico "Ø".

A corrente através de R_L será:

$$I_{RL} = I_{OL} - n\, I_{IL}$$

e a queda de tensão em R_L será:

$$E_{RL} = V_{cc} - V_{OL} \text{ (desejado)},$$

logo:

$$R_L(\text{min}) = \frac{V_{cc} - V_{OL} \text{ (desejado)}}{I_{OL} - n\, I_{IL}} \qquad (2)$$

Se utilizarmos um valor de R_L menor que o calculado pela expressão (2), a corrente fornecida à saída que estiver em "Ø" será maior que o máximo permitido. No circuito da figura 3.21, temos n=3, I_{OL} = 16 mA e I_{IL} = 1,6 mA, de modo que para um V_{OL} (desejado) de Ø,4V temos:

$$R_L(\text{min}) = \frac{5 - 0,4}{16 - 3x1,6} = 0,411\ K\Omega$$

Assim, para a ligação WIRE-AND das figuras 3.20 e 3.21 devemos escolher

$$0,411K\Omega < R_L < 2,989K\Omega$$

para que o circuito funcione corretamente com $V_{OH} \geq 2,4V$ e $V_{OL} \leq 0,4V$.

3.3.5.2 - Saída TRI-STATE

Uma terceira configuração encontrada nos CI'S da família TTL é a chamada saída TRI-STATE.

A figura 3.22 ilustra uma porta NAND TTL com a saída TRI-STATE:

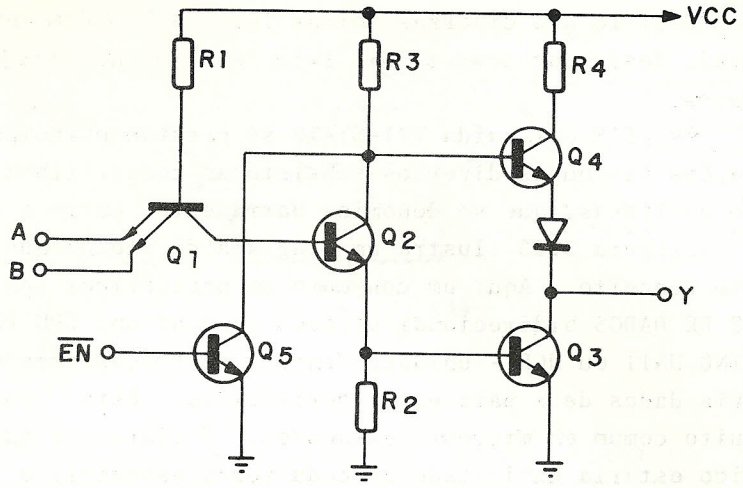


Fig. 3.22 - Porta NAND TTL com a saída TRI-STATE.

Observe que há um transistor adicional, Q_5, cuja condução é controlada por um sinal $\overline{EN}$. A entrada $\overline{EN}$ é uma entrada de habilitação ou ENABLE, e é ativa no estado lógico "Ø", ou seja, o circuito está habilitado quando $\overline{EN}$ = "Ø".

De fato, com $\overline{EN}$ = "Ø" o transistor Q_5 não conduz e sua influência no circuito é desprezível; ele funciona como uma porta NAND comum.

Por outro lado, quando $\overline{EN}$ = "1", o transistor Q_5 conduz e faz com que Q_3 e Q_4 cortem, por não terem praticamente corrente de base. A saída Y vai para um estado de alta impedância (Hi - Z) e é essencialmente transparente para o restante do circuito a que está ligada.

Deste comportamento vem o nome TRI-STATE. Uma saída deste tipo apresenta três estados:

- o estado lógico "Ø",
- o estado lógico "1",
- o estado de alta impedância, no qual a saída está essencialmente desconectada do restante do circuito.

É claro que diversas saídas TRI-STATE podem ser ligadas juntas, desde que *somente uma delas esteja habilitada a cada instante.*

Os CI'S com saída TRI-STATE se prestam principalmente a aplicações nas quais diversos subsistemas compartilham um conjunto de linhas, que se denomina barramento, ou mais comumente BUS. A figura 3.23 ilustra um diagrama de blocos que esclarece este conceito. Aqui um conjunto de periféricos tem acesso a um BUS DE DADOS bidirecional através do qual uma CPU (CENTRAL PROCESSING UNIT ou UCP - Unidade Central de Processamento) recebe e envia dados de e para estes periféricos. Esta configuração é muito comum em microprocessadores. É claro que apenas um periférico estaria habilitado de cada vez e assumiria o controle do BUS DE DADOS.

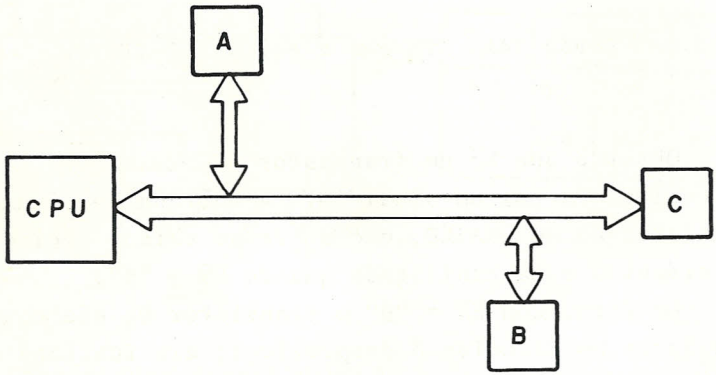


Fig. 3.23 - BUS Compartilhado, Aplicação da Lógica TRI-STATE.

Os CI'S com saída TRI-STATE apresentam velocidade de comutação comparável aos com saída TOTEM-POLE. No entanto há menor número de funções disponíveis com saída TRI-STATE e o custo é mais elevado. Assim, estes CI'S devem ser utilizados somente em aplicações nas quais seu emprego apresenta uma nítida vantagem em relação aos CI'S convencionais, como é o caso da figura 3.23.

3.3.6 - Cuidados no Uso de Circuitos Integrados TTL

3.3.6.1 - Fonte de Alimentação

A fonte de alimentação deve fornecer uma tensão regulada entre 4,75V e 5,25V para todos os pontos do circuito.

A corrente requerida é encontrada somando-se os valores de I_{CC} típicos para cada elemento (dado de manual) e adicionando uma tolerância de 25%, para levar em conta os picos de corrente que ocorrem na comutação entre os estados lógicos. Além disso, deve-se considerar também a corrente exigida pelos resistores de PULL-UP eventualmente ligados às entradas (ver item 3.3.6.3).

Uma forma rápida de estimar a corrente é considerar a dissipação de cada circuito em mW (dado de manual) e dividir a dissipação total pela tensão de alimentação 5V, obtendo a corrente total em mA. Novamente, é necessário adicionar uma margem de segurança.

Nas placas de circuito impresso, as trilhas de V_{CC} e GND devem ser tão largas quanto possível e capacitores de desacoplamento devem ser distribuídos pelas diversas placas. Deve-se usar pequenos capacitores eletrolíticos (10 µF x 6V) e de disco cerâmico (Ø,1 µF x 6V) com os terminais os mais curtos possíveis. Os capacitores cerâmicos são necessários para desacoplar as componentes de alta freqüência geradas pela comutação.

3.3.6.2 - Capacitores de Desacoplamento

Além dos capacitores de desacoplamento mencionados com relação às linhas de alimentação, é importante utilizar outros dentro da placa de circuito impresso.

Conforme já vimos anteriormente, durante a comutação da saída, há um instante em que os dois transistores Q_3 e Q_4 estão conduzindo ao mesmo tempo, acarretando um pico na corrente de alimentação. Isto representa um ruído que pode se propagar pelas linhas de alimentação e causar problemas em outras partes

do circuito. Para evitar que isto ocorra, recomenda-se a colocação de capacitores de disco cerâmico com valor de Ø,Ø1 µF a Ø,1 µF entre V_{CC} e terra de um mesmo CI, assim distribuídos:

- um capacitor para cada quatro CI'S, contendo exclusivamente portas;
- um capacitor para cada dois CI'S MSI (contadores, decodificadores, etc);
- um capacitor para cada CI, situado a mais de 7,5 cm do mais próximo capacitor de desacoplamento.

Estes capacitores devem ter os terminais tão curtos quanto possível e seu uso é muito importante para o correto funcionamento do circuito.

Atenção: não vale somar as capacitâncias necessárias e concentrá-las em um único capacitor, pois o que importa neste caso não é tanto o valor da capacitância mas a maneira como ela está distribuída dentro da placa.

3.3.6.3 - Entradas Não-Usadas

Embora uma entrada aberta represente um nível lógico "1" para TTL, *nunca* se devem deixar abertas as entradas não usadas de um circuito integrado, porque isto torna o circuito mais suscetível a ruídos. Entradas não usadas devem ser conectadas a um nível lógico definido.

Em ordem de preferência são estas as soluções recomendadas para portas AND ou NAND:

a) conectar as entradas não usadas a entradas usadas, na mesma porta; para cálculo do FAN-OUT, duas entradas ligadas juntas representam um FAN-IN de "1" no estado lógico "Ø" e 1,5 no estado lógico "1";

b) conectar as entradas não-usadas à saída de uma porta que permanece sempre no estado lógico "1", o FAN-OUT desta saída pode ser excedido;

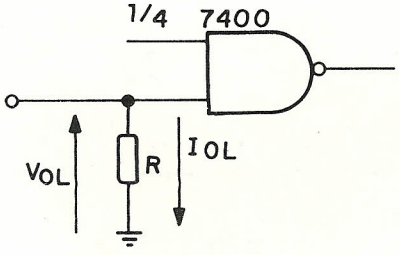
c) conectar as entradas não usadas à V_{CC} através de um resistor de 1 KΩ; de uma a 25 entradas podem ser ligadas a um mesmo re

sistor. Não se recomenda a ligação direta a V_{CC} porque podem ocorrer transientes na fonte que danifiquem o CI.

Para portas OR ou NOR as entradas não utilizadas devem ser aterradas.

3.3.6.4 - Outros Cuidados

Quando uma entrada TTL está no estado lógico "Ø" flui uma corrente I_{OL} que deve ser absorvida pelo circuito externo. Se, por alguma razão, for necessário conectar um resistor entre a entrada e o terra o valor deste deve ser dimensionado de modo que a queda de tensão sobre ele seja $\leq V_{IL}$. No circuito da figura 3.24, por exemplo, R $\leq$ 5ØØ Ω:



$$R \leq \frac{V_{IL}}{I_{IL}} = \frac{0,8}{1,6} = 0,5\ K\Omega$$

Fig. 3.24 - Resistor Conectado entre uma Entrada e o Terra.

Nos circuitos integrados TTL mais novos, há diodos de amortecimento conectados às entradas para evitar oscilações de alta freqüência, conforme ilustra a figura 3.25:

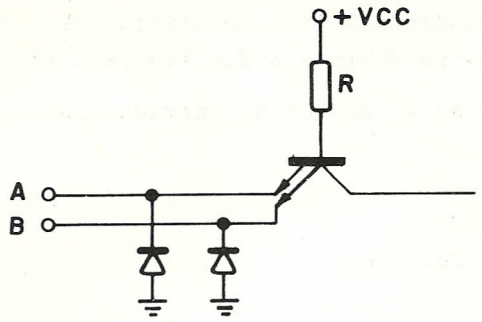


Fig. 3.25 - Diodos de Amortecimento na Entrada.

Se uma entrada se tornar mais de Ø,6V negativa com relação ao terra, ocorrerá a imediata destruição do circuito integrado, a menos que a corrente seja limitada a, no máximo, 1Ø mA.

3.4 - Família CMOS

3.4.1 - Visão Geral

Os circuitos integrados da família CMOS são encontrados em duas séries principais: a série 54C/74C e a série 4ØØØ.

A série 54C/74C engloba circuitos integrados que correspondem diretamente a seus homônimos da série 54/74 TTL. Assim, por exemplo, o CI 74CØØ é a versão C-MOS do CI 74ØØ TTL. Os CI'S do tipo 54C operam na faixa de temperatura de -55°C a +125°C, ao passo que os do tipo 74C operam em temperaturas de -40°C a +85°C.

A série 4ØØØ é encontrada em duas versões: 4ØØØA (standard) e 4ØØØB (BUFFERED) sendo a versão 4ØØØB mais recente. Os CI'S da versão 4ØØØA operam entre -55°C e +125°C (4ØØØA-2) ou -40°C e +85°C (4ØØØA-9), enquanto os da versão 4ØØØB operam entre -55°C e +125°C.

Os circuitos integrados da família CMOS operam com ten

sões entre 3 e 15V, com exceção dos do tipo 4ØØØB, que operam com tensões entre 3 e 18V.

3.4.2 - Princípios Básicos

Os elementos básicos dos circuitos integrados CMOS são transistores de efeito de campo do tipo MOSFET (METAL OXIDE SEMICONDUCTOR FIELD EFFECTT TRANSISTOR) ENHACEMENT de canal N e canal P.

A figura 3.26 ilustra a constituição de um MOSFET ENHACEMENT de canal N e o símbolo utilizado para representá-lo.

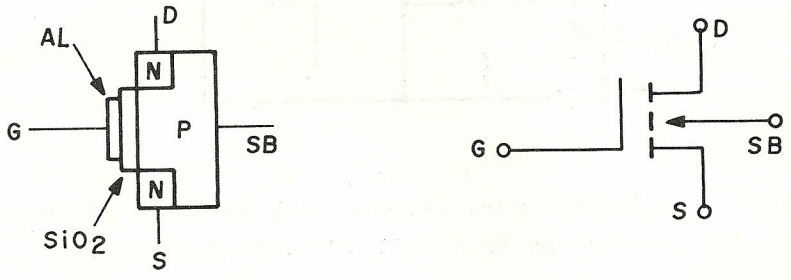


Fig. 3.26 - Diagrama do MOSFET ENHACEMENT de Canal N e seu símbolo.

Os quatro terminais são:

S - Source (supridouro),

D - Drain (dreno),

G - Gate (porta),

SB - Substrate (substrato).

Conforme se vê pelo diagrama da figura 3.26, no MOSFET de canal N, utiliza-se um substrato de material P no qual são difundidas as duas regiões N que formam o supridouro e o dreno; a porta é isolada do substrato por uma camada de óxido, de modo que ela é um circuito aberto (resistência da ordem de $1Ø^{9}\Omega$) para

todos os efeitos práticos.

Vamos analisar o funcionamento do MOSFET ENHACEMENT de canal N através do circuito mostrado na figura 3.27; o substrato é geralmente conectado ao supridouro como indicado.

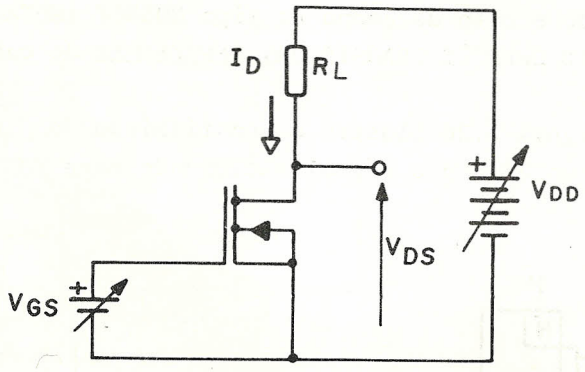


Fig. 3.27 - Circuito para Levantamento da Curva de Transferência de um MOSFET ENHACEMENT de Canal N.

Façamos inicialmente V_{GS} e V_{DD} iguais a zero. Aumentando V_{DD}, veremos que o transistor não conduz praticamente corrente alguma, pois a região P entre o supridouro e o dreno apresenta uma alta resistência. É o estado de corte do MOSFET.

Repetindo a experiência com diversos valores de V_{GS}, veremos que para $V_{GS} = V_T$ (tensão de limiar, ou THRESHOLD) o transistor começa a conduzir e a corrente I_D passa a variar com a variação de V_{DS}. A condução deve-se ao fato de que para $V_{GS} \geq V_T$ ocorre a formação de uma camada de material N abaixo da porta (camada de inversão), conectando a fonte e o dreno e permitindo a circulação de corrente. Para diversos valores de $V_{GS} > V_T$, obteremos um conjunto de curvas de transferência de I_D em função de V_{DS} que terão o aspecto ilustrado na figura 3.28.

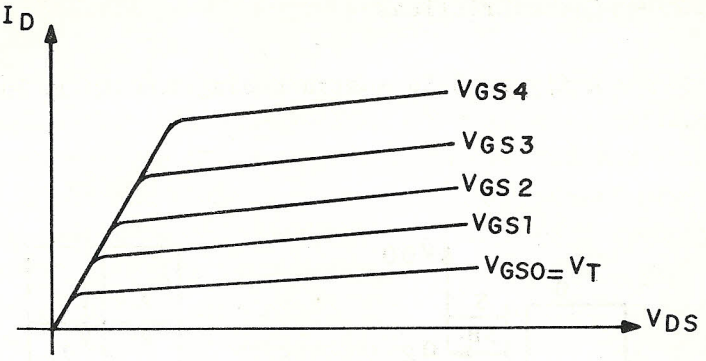


Fig. 3.28 - Curvas características de I_D X V_{DS} para o MOSFET de Canal N.

O valor de V_T depende da construção do transistor e é geralmente da ordem de 1 a 3V. Na curva característica I_D X V_{DS} distinguimos duas regiões: uma região linear, na qual I_D aumenta com o aumento de V_{DS} e uma região de saturação na qual I_D é praticamente constante e independente de V_{DS}. Quando o MOSFET é utilizado em um circuito lógico, ele é operado ou no corte ($V_{GS} < V_T$) ou na saturação ($V_{GS} > V_T$ e V_{DS} suficientemente alto).

O comportamento do MOSFET ENHACEMENT de canal P, cujo diagrama e símbolo, estão representados na figura 3.29 é análogo, exceto que as polaridades são invertidas (V_T e V_{DS} são negativos).

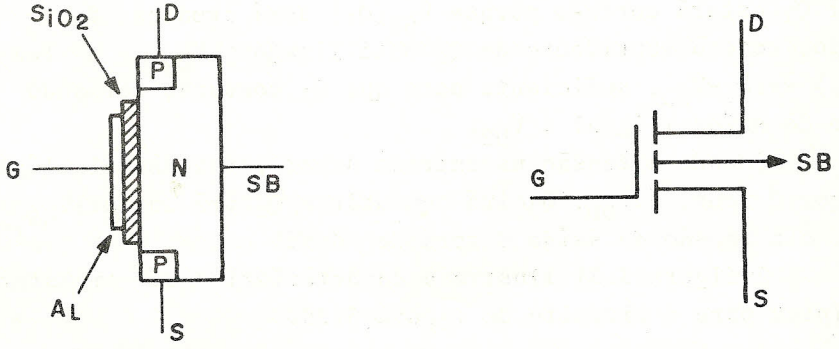
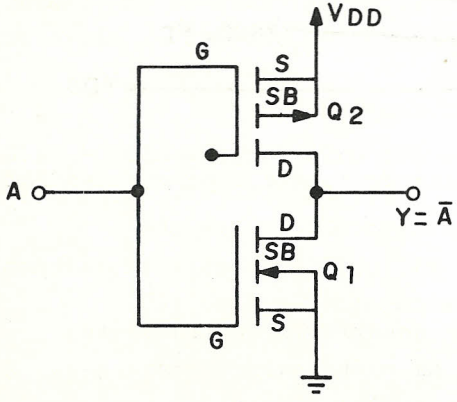


Fig. 3.29 - Diagrama e Símbolo do MOSFET ENHACEMENT de Canal P.

3.4.3 - Características

A figura 3.30 mostra o diagrama de um inversor lógico CMOS.



A	y
ØV	V_{DD}
V_{DD}	ØV

Fig. 3.30 - Inversor Lógico CMOS e sua Tabela de Estados.

Observe atentamente a figura 3.30 e veja como estão ligados os transistores Q_1 e Q_2. Q_1 é um MOSFET ENHACEMENT de canal N e Q_2 um MOSFET ENHACEMENT de canal P.

Suponhamos que a entrada A esteja aterrada, o transistor Q_1 estará cortado porque $V_{GS}(Q_1)$ será igual a Ø V. No entanto, como o supridouro de Q_2 está ligado a V_{DD}, a tensão $V_{GS}(Q_2)$ será $-V_{DD}$, suficiente para que Q_2 conduza, tornando a tensão de saída Y igual a V_{DD}.

Quando a tensão na entrada A estiver no nível lógico "1", que é igual a V_{DD}, Q_1 irá conduzir e Q_2 irá cortar($V_{GS}(Q_2)$ = ØV), e a tensão de saída Y será de Ø V.

A figura 3.31 ilustra a característica de transferência típica para o circuito da figura 3.30.

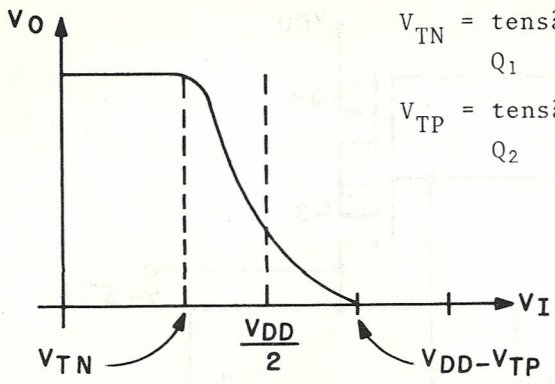


Fig. 3.31 - Característica de Transferência do Inversor C-MOS.

Dois pontos podem ser destacados na operação do circuito da figura 3.30.

Em primeiro lugar, observamos que a entrada é um circuito aberto e não consome praticamente nenhuma corrente do circuito que a alimenta. Isto significa que uma saída C-MOS pode alimentar um número muito grande de entradas.

De fato, a limitação decorre da velocidade de comutação necessária, e não de considerações de corrente. Cada entrada representa uma pequena capacitância e introduz um atraso da ordem de 3ns na comutação e é este o fator que limita o número de entradas que se pode ligar a uma saída.

O outro ponto digno de nota é o fato de que entre V_{DD} e terra há sempre um transistor cortado, apresentando uma altíssima impedância. Assim, o consumo de potência é muito reduzido, da ordem de apenas 1Ø nW. Esta característica torna os circuitos integrados C-MOS especialmente indicados nas aplicações onde o baixo consumo seja essencial, como é o caso de equipamentos operados com baterias.

A figura 3.32 mostra uma porta NOR C-MOS.

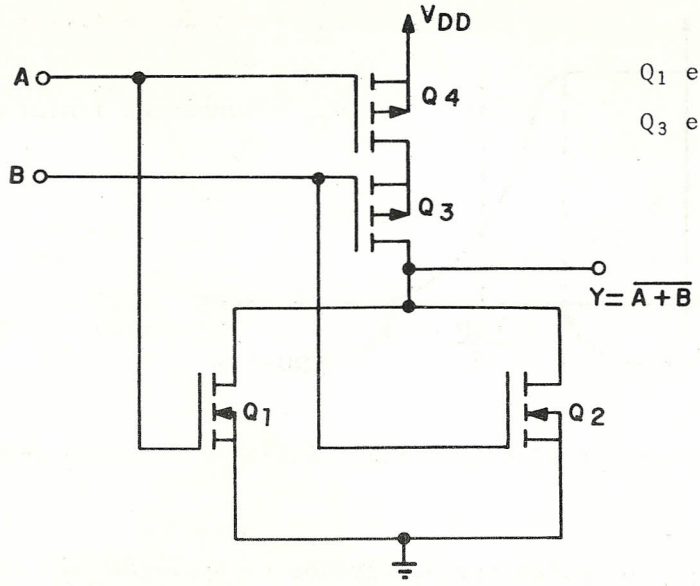
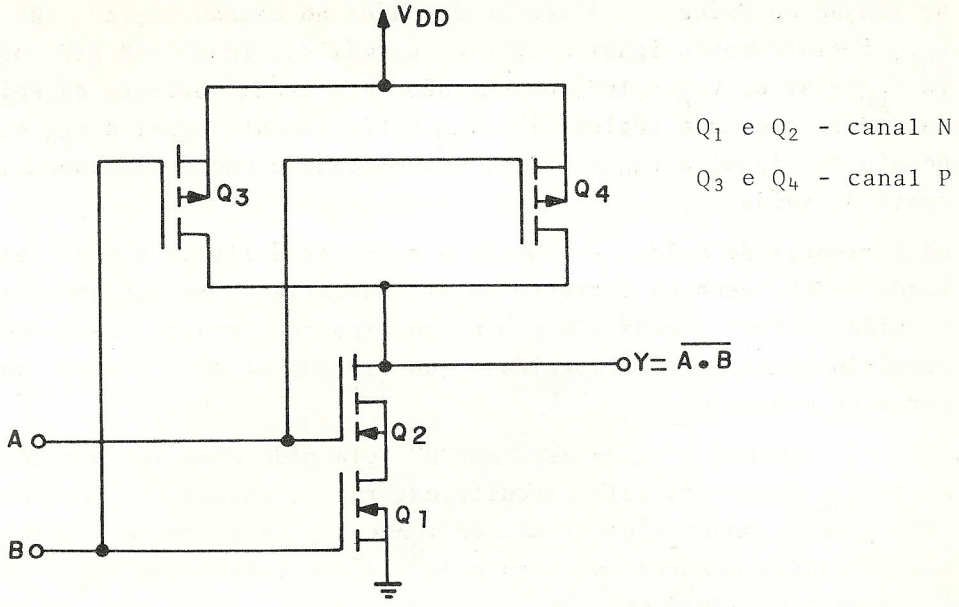


Fig. 3.32 - Portas NOR C-MOS.

Neste circuito vemos que se uma (ou ambas) das entradas estiverem no nível lógico "1" um dos transistores (ou ambos) de canal N conduzirão, levando a saída para o nível lógico "Ø". Somente quando as duas entradas estiverem em "Ø", os dois transistores de canal P conduzirão simultaneamente, levando a saída para "1".

Na figura 3.33, mostra-se o esquema da porta NAND C-MOS.



Q_1 e Q_2 - canal N

Q_3 e Q_4 - canal P

Fig. 3.33 - Porta NAND C-MOS.

Neste caso, se uma (ou ambas) das entradas estiverem em "Ø" , um (ou ambos) dos transistores de canal P estarão conduzindo e a saída ficará em "1". Somente quando as duas entradas estiverem em "1", os dois transistores de canal N conduzirão ao mesmo tempo, levando a saída para "Ø".

Vamos, agora, analisar alguns parâmetros dos circuitos integrados C-MOS, conforme fornecidos pelo manual. Tomaremos como exemplo o CI 4ØØ1, que contém quatro portas NOR de duas entradas (QUAD 2-INPUT POSITIVE NOR GATE). Observe que muitos parâmetros são especificados para dois valores de V_{DD} (V_{DD} = 5V e V_{DD} = 1ØV) e que os valores apresentados se referem à operação em 25°C.

a) Corrente de Fuga de Entrada - Conforme já vimos anteriormente, a entrada de um circuito integrado C-MOS é praticamente um circuito aberto. As correntes I_{IL} e I_{IH}, que são as correntes de fuga de entrada nos estados lógicos "Ø" e "1", respectivamente, são de 1Ø pA para V_{DD} = 1ØV.

b) Tensão de Saída - A tensão de saída no estado lógico "Ø", V_{OL}, é tipicamente igual a Ø V e, no máximo, igual a Ø,ØlV para V_{DD} = 5V ou V_{DD} = lØV, quando não se consome corrente da saída. Para o estado lógico "1", V_{OH} é tipicamente igual a V_{DD} e, no mínimo, igual a V_{DD} - Ø,ØlV, também quando não se consome corrente da saída.

c) Correntes de Saída - Quando a saída está ligada a uma entrada C-MOS, nenhuma corrente estará envolvida. No entanto, se a saída estiver ligada a algum outro tipo de circuito, será necessário considerar a capacidade que ela possui de drenar ou fornecer corrente.

Quando a saída está em "Ø" ela pode absorver uma corrente I_{OL} fornecida pelo circuito externo,e, quando no estado "1", a saída pode fornecer uma corrente I_{OH} ao circuito externo. A tabela mostrada na figura 3.34 dá os valores de I_{OL} e I_{OH} para o CI 4ØØ1 em 25^0C:

	MIN	TIP	MAX	UNIDADE	V_{DD}	CONDIÇÕES
I_{OL}	Ø,4	1	-	mA	5	V_{OL} = Ø,4V
	Ø,9	2,5	-	mA	lØ	V_{OL} = Ø,5V
I_{OH}	-Ø,5	-2	-	mA	5	V_{OH} = 2,5V
	-Ø,5	-1	-	mA	lØ	V_{OH} = 9,5V

Fig. 3.34 - Correntes de Saída para o CI 4ØØ1 CMOS.

d) Consumo de Potência - A corrente quiescente consumida da fonte de alimentação I_{DD} é muito baixa, conforme já indicamos anteriormente. No manual, encontramos para I_{DD} um valor típico de Ø,ØØ1 μA, e o máximo de Ø,Ø5 μA para V_{DD} = 5V e 0,001 μA para V_{DD} = lØV. Isto corresponde a uma dissipação de potência típica de 5 nW em 5V e lØ nW em lØV. Estes valores são quiescentes, ou seja, correspondem a entradas e saídas fixas.

Quando as entradas variam, a saída da porta muda de estado e durante a comutação há um período no qual os transisto

res conduzem ao mesmo tempo, aumentando a corrente e a dissipação de potência. Assim, quanto mais rapidamente variarem as entradas, tanto maior será o consumo. A figura 3.35 mostra a dissipação de potência em função da freqüência de comutação para o circuito integrado. Mesmo com freqüências de comutação de vários MHZ, o consumo de potência ainda é relativamente baixo.

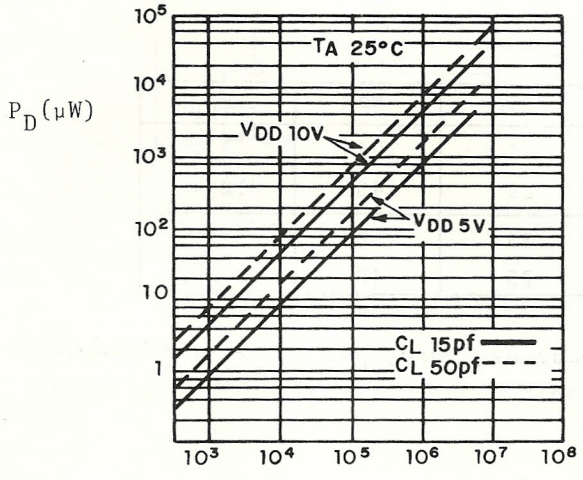
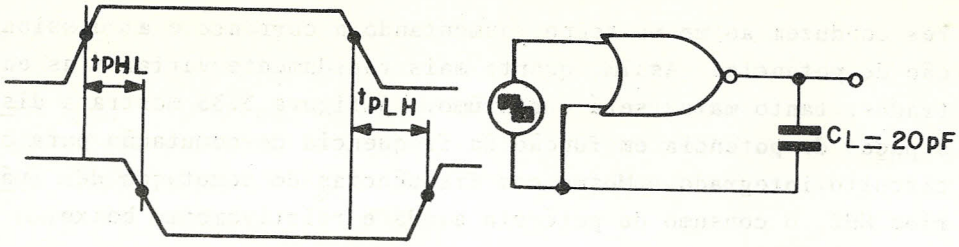


Fig. 3.35 - Dissipação de Potência Típica X Freqüência do Sinal de Entrada.

e) Atraso de Propação - Os atrasos t_{pLH} e t_{pHL} são medidos conforme indica a figura 3.36, considerando uma capacitância C_L de 2Ø pF na saída. A tabela mostrada na figura 3.36 indica que o circuito é mais veloz quando operado com maior tensão de alimentação. Conforme já dissemos anteriormente, a ligação de entradas C-MOS a uma saída, aumenta a capacitância e, portanto, o atraso. Pode-se estimar que cada entrada representa um atraso adicional de 3 ns e uma capacitância de 5 pF.



	Tip	Max	V_{DD}
t_{pLH} ns	35 25	95 45	5 V 1Ø V
t_{pHL} (su)	35 25	5Ø 4Ø	5 V 1Ø V

Fig. 3.36 - Atraso de Propagação e Características de Chaveamento.

Concluímos também que a velocidade de comutação dos circuitos integrados C-MOS é inferior a dos circuitos integrados TTL.

f) Imunidade ao Ruído - Os circuitos integrados CMOS apresentam excelente imunidade a ruídos. A margem de ruído *garantida* é de 30% da tensão de alimentação e a típica é de 45%. Assim, para um circuito operado a partir de uma tensão V_{DD}=5V, podemos ter certeza de que qualquer tensão de entrada entre Ø e 1,5Ø V será interpretado como nível lógico "Ø" e qualquer tensão de entrada entre 3,5 e 5,Ø V será interpretado como nível lógico "1". Para V_{DD} = 1Ø V, os valores seriam:

- Ø a 3V → nível lógico "Ø"
- 7 a 1Ø V → nível lógico "1".

Esta grande imunidade ao ruído torna os circuitos integrados C-MOS indicados para aplicações em ambientes com alto

nível de ruído elétrico, como é o caso de ambientes industriais.

3.4.4 - Cuidados no Uso de Circuitos Integrados C-MOS

3.4.4.1 - Fonte de Alimentação

Os circuitos integrados C-MOS podem operar com tensões de alimentação de 3 a 15 V ou 3 a 18 V, e seu consumo é extremamente baixo. Assim, a fonte de alimentação pode ser mais simples e compacta e com pior regulação do que a requerida para um circuito comparável implementado com CI'S TTL.

É mais vantajoso operar com tensões de alimentação na faixa de 9 a 12 V porque a velocidade, a capacidade de fornecer ou absorver corrente e a imunidade ao ruído são maiores, conforme indica a análise destas características, feita no item anterior.

3.4.4.2 - Entradas Não Usadas

É absolutamente essencial que todas as entradas não utilizadas sejam ligadas a uma tensão definida (V_{DD} ou terra, conforme o caso). Uma entrada flutuante assumirá um estado lógico indefinido, tornando a operação do circuito imprevisível. Pode, ainda, acontecer que a entrada aberta faça o transistor correspondente ficar na região ativa, aumentando muito a corrente consumida pelo CI.

3.4.4.3 - Eletricidade Estática

As entradas de um circuito integrado C-MOS são essencialmente circuitos abertos. O acúmulo de cargas elétricas na região do GATE pode dar origem a um campo elétrico muito grande, causando a perfuração do dielétrico e a destruição do CI.

Em geral as entradas contêm circuitos de proteção com diodos, como os ilustrados na figura 3.37 que se destinam a evitar que a tensão atinja valores capazes de danificar o CI.

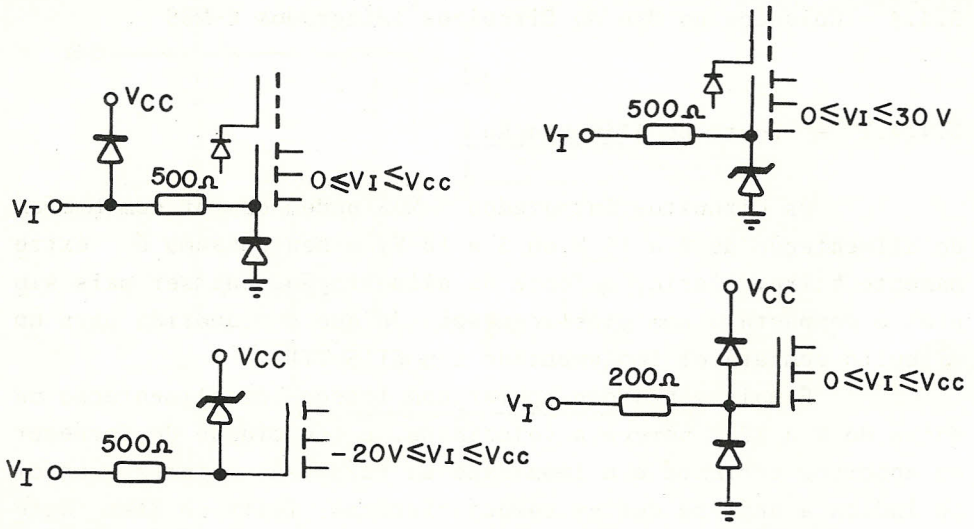


Fig. 3.37 - Circuitos de Proteção Internos em Entradas CMOS.

Mesmo assim, recomenda-se que algumas precauções sejam tomadas no manuseio dos CI'S da família C-MOS. Quando não em uso, os CI'S devem ser fixados em uma tira de espuma condutora e ao segurá-los deve-se evitar fazê-lo pelos pinos. Todo o equipamento deve ser aterrado e o operador deve usar uma pulseira metálica conectada ao terra por um resistor de 1 MΩ.

Os circuitos de proteção internos ao CI requerem que as tensões de entrada permaneçam nos limites especificados na figura 3.37. Se estes limites tiverem que ser excedidos, a corrente tem que ser limitada a, no máximo, 1Ø mA ou ocorrerá a destruição do circuito integrado.

3.5 - Interface TTL/C-MOS e C-MOS/TTL

Quando circuitos integrados das famílias TTL e C-MOS

necessitam ser interligados, devemos considerar as peculiaridades de cada caso.

Em primeiro lugar, vejamos o que ocorre quando uma saída TTL deve alimentar entradas C-MOS. Se todo o circuito utilizar uma alimentação de 5V, qualquer CI TTL poderá ser usado para alimentar a entrada C-MOS. Como as entradas C-MOS não consomem corrente, não há nenhum problema quanto a este aspecto. No estado lógico "Ø", a saída TTL apresentará $V_{OL} \leq Ø,4V$, que o circuito C-MOS entenderá como nível "Ø". No entanto, para o nível lógico "1", a saída TTL apresentará uma tensão V_{OH} típica de cerca de 3,3V, enquanto o circuito C-MOS requer uma tensão $\geq 3,5V$ na entrada para caracterizar o nível lógico "1". Torna-se necessário adicionar um resistor de PULL-UP externo entre a saída TTL e V_{CC}, de modo a aumentar a tensão de saída. O valor deste resistor não é crítico, devendo-se apenas não fazê-lo muito pequeno, para evitar uma corrente excessiva no estado "Ø". Um valor comum e 22 kΩ, como indica a figura 3.38

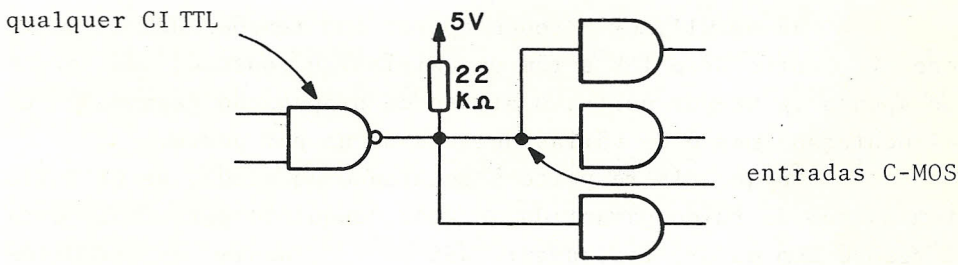


Fig. 3.38 - Interface TTL/CMOS para Alimentação de 5V.

Pode ocorrer que os circuitos C-MOS sejam alimentados com uma tensão maior que 5V, por exemplo 1ØV. Neste caso, o CI TTL que os alimenta deve ser do tipo BUFFER OPEN-COLLECTOR(7406, 7407), com um resistor de PULL-UP externo de valor conveniente conectado à tensão mais elevada.

Finalmente, quando um CI C-MOS alimenta uma entrada TTL, devemos considerar as correntes envolvidas. A saída C-MOS

pode drenar Ø,4 mA no estado baixo e fornecer Ø,5 mA no estado alto. Uma entrada TTL normal fornece uma corrente de 1,6 mA no estado baixo, logo a saída C-MOS não é capaz de absorver a corrente requerida. Já uma entrada TTL da versão LOW POWER fornece Ø,18 mA no estado baixo e, assim, a saída C-MOS pode drenar a corrente de até duas entradas TTL-L. Para interface com TTL normal, há BUFFERS CMOS (4Ø49,4Ø5Ø) com maior capacidade de absorção de corrente.

3.6 - Comparação entre TTL e CMOS

Do exposto anteriormente podemos concluir fazendo uma comparação entre os circuitos integrados TTL e CMOS:

Os CI'S TTL operam a partir de uma alimentação V_{cc} de 5 ± Ø,25V que deve ser bem regulada e consomem uma potência relativamente elevada da ordem de 1Ø mW por porta lógica; apresentam altas velocidades de comutação, com um atraso da ordem de apenas 1Ø ns por porta.

Já os CI'S CMOS podem operar com tensões de alimentação V_{cc} entre 3V e 15V e têm um baixíssimo consumo, da ordem de apenas 1Ø nW por porta, o atraso de propagação depende da alimentação, mas é de várias dezenas de ns por porta.

No que diz respeito à imunidade ao ruído, os CI'S TTL têm margem de ruído garantida de Ø,4V, enquanto para CMOS esta margem é bem maior, tipicamente 45% de V_{cc}; assim, os circuitos integrados CMOS se prestam a aplicações em ambientes com alto nível de ruído elétrico.

Para TTL o FAN-OUT é de 1Ø, limitando a este valor o número de entradas que podem ser conectadas a uma saída; para CMOS o FAN-OUT é, essencialmente, infinito e o fator que limita o número de entradas que pode ser ligado a uma saída é a velocidade de comutação.

CAPÍTULO 4 - PORTAS LÓGICAS E CIRCUITOS COMBINACIONAIS

4.1 - Introdução

De maneira bastante ampla, podemos classificar os circuitos digitais em combinacionais e seqüenciais.

Circuitos combinacionais são aqueles nos quais a saída depende apenas do estado presente das entradas; são constituídos fundamentalmente de portas lógicas.

Circuitos seqüenciais são aqueles nos quais a saída depende não só do estado presente das entradas, mas também de seus estados anteriores; são constituídos fundamentalmente de portas lógicas e FLIP FLOPS.

Neste capítulo, abordaremos os problemas de análise e e síntese (projeto) de circuitos combinacionais, utilizando circuitos integrados das famílias TTL e CMOS para implementá-los.

4.2 - Portas Lógicas da Família TTL

Há uma grande variedade de portas lógicas disponíveis na série 74TTL. Neste item serão apresentados diversos tipos de portas desta série, acompanhadas de uma breve descrição. Os parâmetros indicados para cada porta referem-se à versão normal ou standard e são valores típicos fornecidos pelos manuais. Esta apresentação não é, de forma alguma, exaustiva e se restringe aos CI'S utilizados em exemplos ao longo deste e de outros capítulos; o manual deve ser consultado para uma lista completa de todas as funções disponíveis e suas características.

7400 - QUAD 2-INPUT NAND GATE

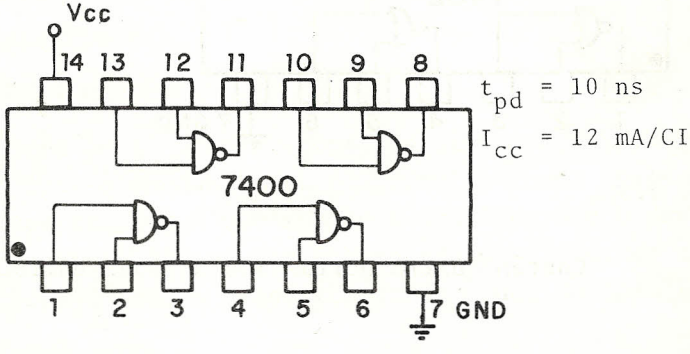
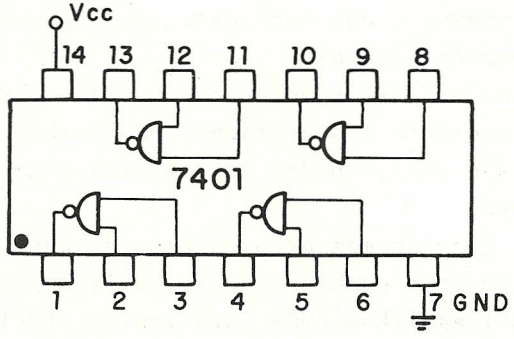


Fig. 4.1

Este CI contém 4 (QUAD) portas NAND de duas entradas (2-INPUT NAND GATE); as quatro portas NAND são independentes.

7401 - QUAD 2-INPUT NAND GATE
(SAÍDA OPEN-COLLECTOR)

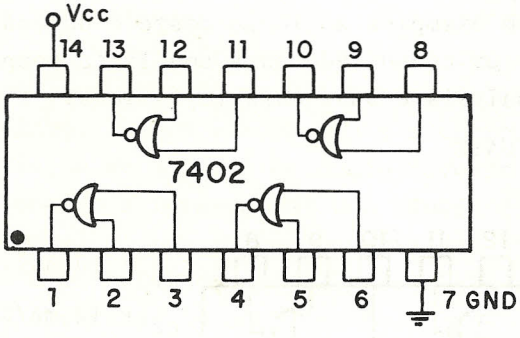


t_{PHL} = 8ns
t_{PLH} = 35ns
I_{cc} = 8mA/CI

Fig. 4.2

Contém quatro portas NAND de duas entradas com a saída em coletor aberto; para que uma saída possa atingir o estado lógico "1" é necessária a conexão de um resistor externo para V_{cc} .

7402 - QUAD 2-INPUT NOR GATE

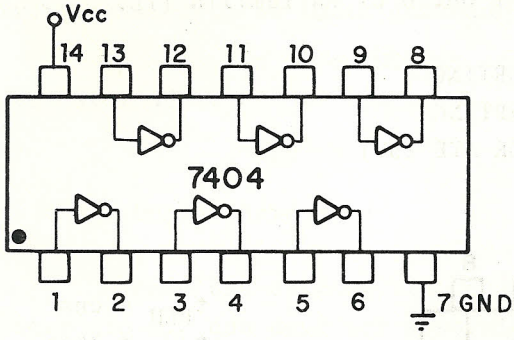


t_{pd} = 10ns
I_{cc} = 12mA/CI

Fig. 4.3

Contém quatro portas NOR de duas entradas.

7404 - HEX INVERTER

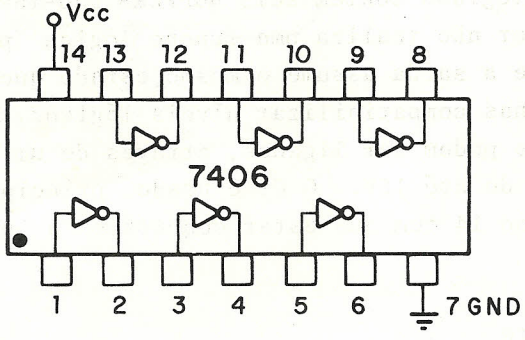


t_{pd} = 10ns
I_{cc} = 12mA/CI

Fig. 4.4

Contém seis (HEX) inversores lógicos.

7406 - HEX BUFFER, INVERTING
HEX DRIVER, INVERTING
(SAÍDA OPEN-COLLECTOR ATÉ 15V)



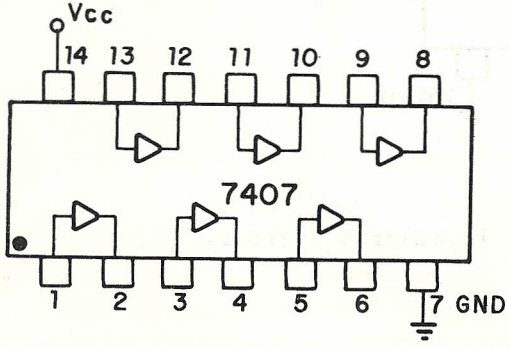
t_{PLH} = 15ns
t_{PHL} = 10ns
I_{cc} = 30mA/CI
I_{OL} = 40mA

Fig. 4.5

Este CI contém seis BUFFERS (isoladores) ou DRIVERS inversores. Denomina-se BUFFER ou DRIVER ao circuito que possui a característica de suportar tensões e fornecer ou drenar correntes mais elevadas que os CI'S normais da série; estes circuitos são utilizados principalmente em interfaces. As saídas são em coletor aberto e o resistor externo pode estar ligado a uma tensão de até 15 V; uma saída no estado baixo é capaz de drenar até 40 mA (contra os 16 mA para os CI'S normais). É im

portante destacar que o pino 14 deve estar ligado ao V_{cc} = = 5,0 ± 0,25V, como qualquer outro CI da família TTL.

7407 - HEX BUFFER, NON-INVERTING
HEX DRIVER, NON-INVERTING
(SAÍDA OPEN-COLLECTOR ATÉ 15V)

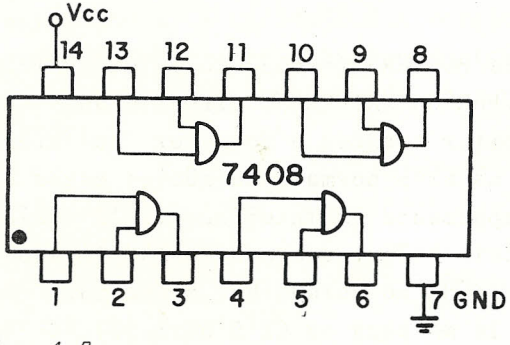


t_{PLH} = 6ns
t_{PHL} = 20ns
I_{cc} = 25 mA/CI
I_{OL} = 40 mA

Fig. 4.6

Este circuito integrado contém seis BUFFERS não-inversores; o BUFFER não-inversor não realiza uma função lógica propriamente dita, uma vez que a saída assume o mesmo estado que a entrada. Sua função é apenas compatibilizar níveis lógicos, pois as saídas em coletor aberto podem ser ligadas, através de um resistor externo, a tensões de até 15V. O CI é usado principalmente em interfaces. O pino 14 tem que estar conectado a uma tensão V_{cc} = 5±0,25V.

7408 - QUAD 2-INPUT AND GATE

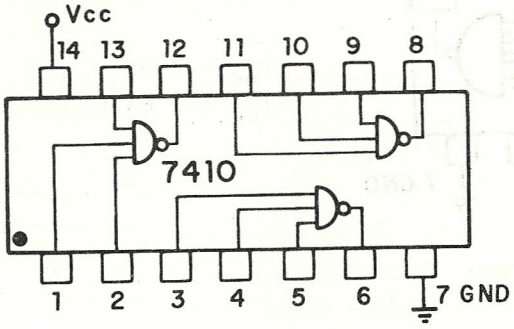


t_{pd} = 15ns
I_{cc} = 15 mA/CI

Fig. 4.7

Contém quatro portas AND de duas entradas.

7410 - TRIPLE 3-INPUT NAND GATE

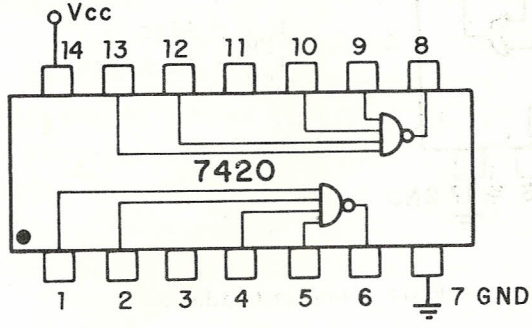


t_{pd} = 9ns
I_{cc} = 6 mA/CI

Fig. 4.8

Contém três (TRIPLE) portas NAND de três entradas.

7420 - DUAL 4-INPUT NAND GATE

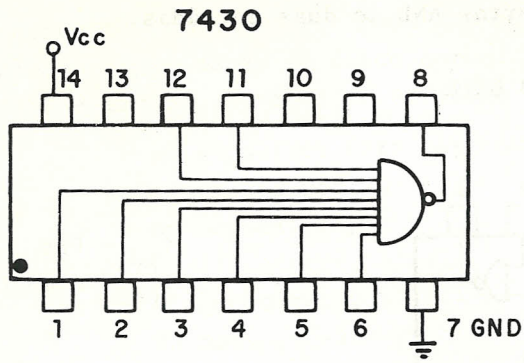


t_{pd} = 10ns
I_{cc} = 4 mA/CI

Fig. 4.9

Contém duas (DUAL) portas NAND de quatro entradas.

7430 - 8-INPUT NAND GATE

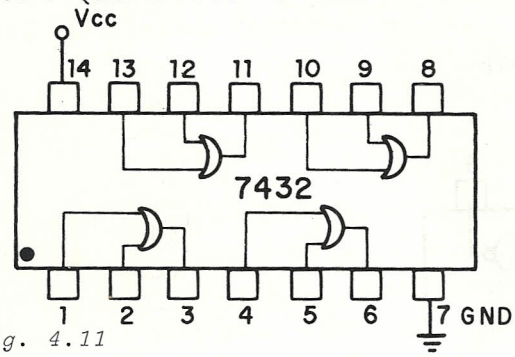


t_{pd} = 10ns
I_{cc} = 2 mA/CI

Fig. 4.10

Contém uma porta NAND de 8 entradas.

7432 - QUAD 2-INPUT OR GATE



t_{pd} = 12ns
I_{cc} = 19 mA/CI

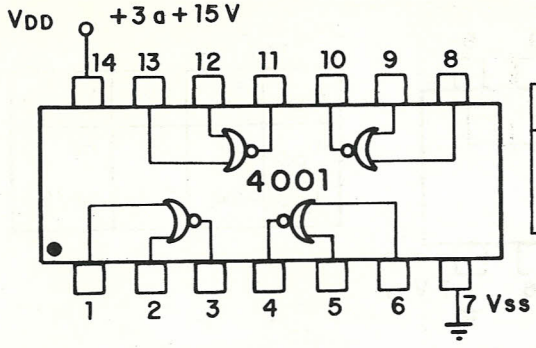
Fig. 4.11

Contém quatro portas OR de duas entradas.

4.3 - Portas Lógicas da Família C-MOS

Neste item serão apresentadas diversas portas lógicas da série 4000 CMOS, acompanhadas de uma breve descrição. Os parâmetros indicados são valores típicos fornecidos pelos manuais; a apresentação se restringe também aos CI'S utilizados em exemplos neste e em outros capítulos. O manual deve ser consultado para funções adicionais aqui não incluidas.

4001 - QUAD 2-INPUT NOR GATE

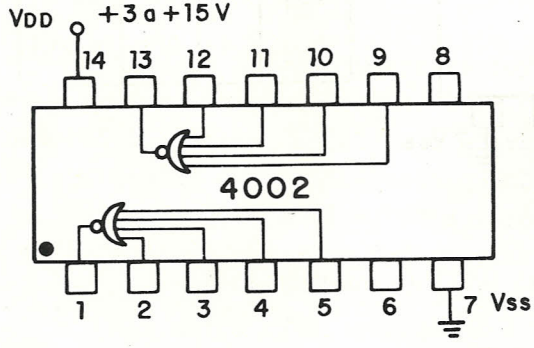


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	0,4	0,8	mA/CI

Fig. 4.12

Contém quatro portas NOR de duas entradas. Conforme já visto no capítulo 3, as características de velocidade e consumo dependem da tensão de alimentação. O consumo apresentado refere-se a uma frequência de comutação de 1 MHZ.

4002 - DUAL 4-INPUT NOR GATE

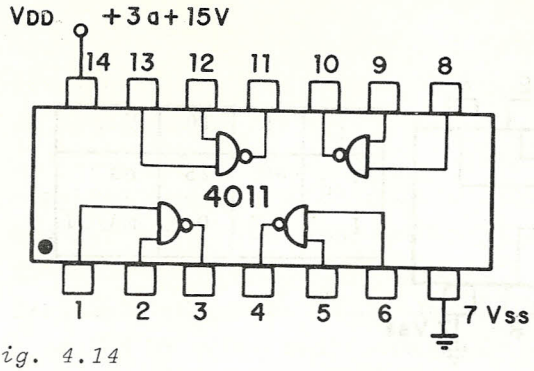


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	0,4	0,8	mA/CI

Fig. 4.13

Contém duas portas NOR de quatro entradas.

4011 - QUAD 2-INPUT NAND GATE

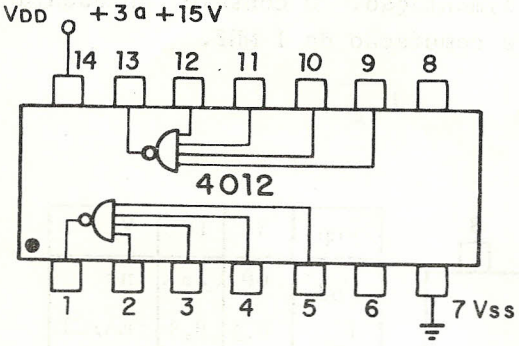


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	0,4	0,8	mA/CI

Fig. 4.14

Contém quatro portas NAND de duas entradas.

4012 - DUAL 4-INPUT NAND GATE

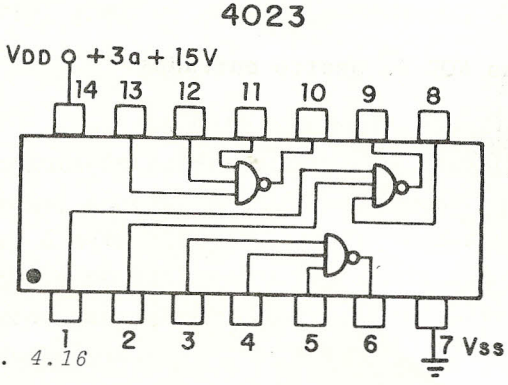


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	0,4	0,8	mA/CI

Fig. 4.15

Contém duas portas NAND de quatro entradas.

4023 - TRIPLE 3-INPUT NAND GATE

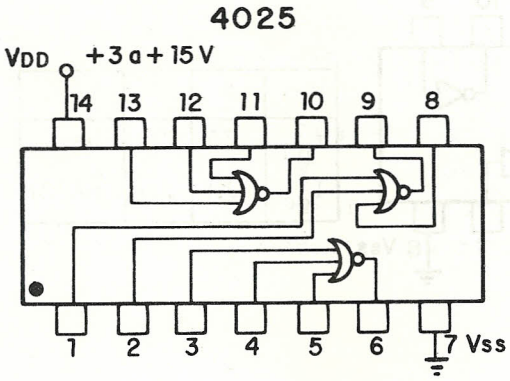


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	0,6	1,2	mA/CI

Fig. 4.16

Contém três portas NAND de três entradas.

4025 - TRIPLE 3-INPUT NOR GATE

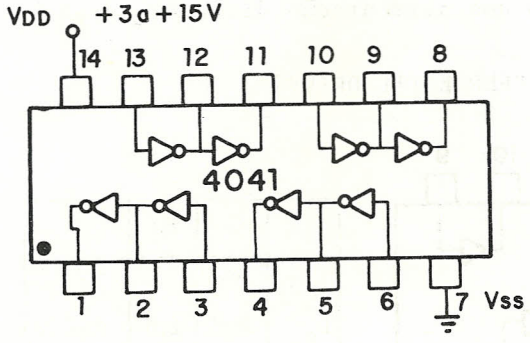


V_{DD}	5	10	V
t_{pd}	60	25	ns
I_{cc}	1,2	2,4	mA/CI

Fig. 4.17

Contém três portas NOR de três entradas.

4041 - QUAD TTL BUFFER



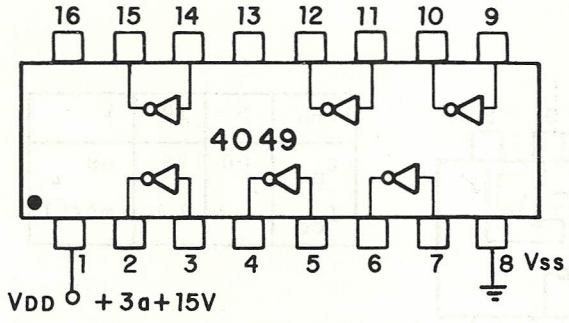
V_{DD}	5	10	V
t_{pd}	75	45	ns
I_{cc}	1,6	3,2	mA/CI

Fig. 4.18

Este CI contém quatro BUFFERS, que possuem saídas inversoras (pinos 2,5,9 e 12) e não inversoras (pinos 1,4,8 e 11). *Quando utilizado com uma alimentação de 5V*, as saídas são compatíveis com TTL; as saídas não inversoras podem drenar até 3,2 mA (duas cargas TTL) e as inversoras até 1,6mA (uma carga

TTL).

4049 - HEX INVERTING BUFFER Ɇ TTL DRIVER

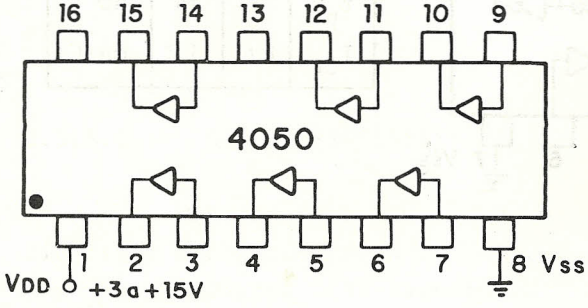


V_{DD}	5	10	V
t_{pd}	35	25	ns
I_{cc}	0,8	1,6	mA/CI

Fig. 4.19

Este CI contém 6 BUFFERS inversores. A tensão aplicada ao pino 1 determina a excursão da saída, mas tensões de entrada de até 15V podem ser aplicadas à entrada, independentemente da tensão no pino 1. Se a tensão no pino 1 é de 5V, a saída é compatível com TTL e pode drenar até 3,2 mA. Do exposto, concluímos que este CI é especialmente indicado para interface entre TTL e CMOS utilizando uma alimentação diferente de 5V.

4050 - HEX NON-INVERTING BUFFER Ɇ TTL DRIVER



V_{DD}	5	10	V
t_{pd}	60	30	ns
I_{cc}	0,8	1,6	mA/CI

Fig. 4.20

Este CI contém seis BUFFERS não inversores e também é especialmente indicado para circuitos de interface com TTL, quan

do a lógica CMOS a ser interligada usa uma alimentação diferente de 5V. Assim como para o 4049, a tensão no pino 1 determina a excursão da saída, podendo a entrada variar entre Ø e 15V, independente da tensão no pino 1. A saída é capaz de drenar até 3,2 mA (duas cargas TTL). A figura 4.21 ilustra a idéia da interface C-MOS/TTL utilizando os CI's 4049 e/ou 4050:

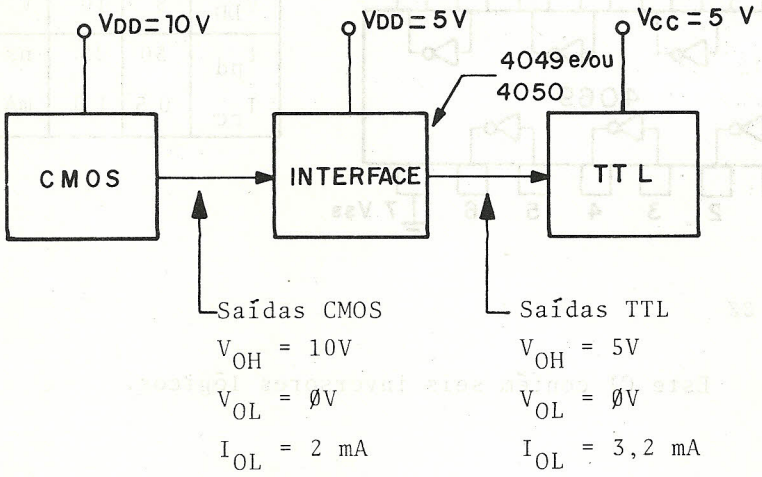
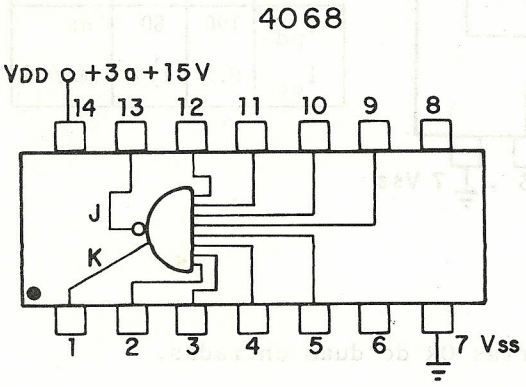


Fig. 4.21 - Interface C-MOS/TTL.

4068 - 8-INPUT NAND GATE

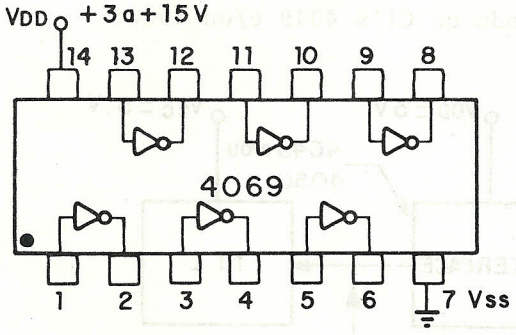


V_{DD}	5	10	V
t_{pd}	325	130	ns
I_{cc}	0,5	1,0	mA/CI

Fig. 4.22

Contém uma porta NAND de oito entradas.

4069 - HEX INVERTER

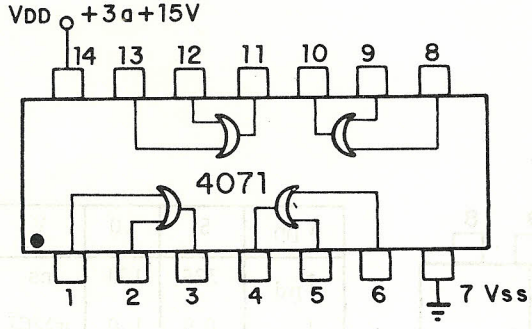


V_{DD}	5	10	V
t_{pd}	50	25	ns
I_{cc}	0,5	1,0	mA/CI

Fig. 4.23

Este CI contém seis inversores lógicos.

4071 - QUAD 2-INPUT OR GATE

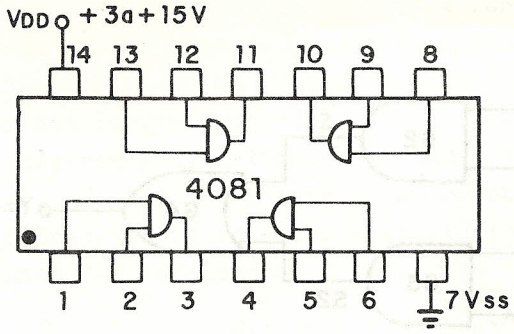


V_{DD}	5	10	V
t_{pd}	190	80	ns
I_{cc}	0,5	1,0	mA/CI

Fig. 4.24

Contém quatro portas OR de duas entradas.

4081 - QUAD 2-INPUT AND GATE



V_{DD}	5	10	V
t_{pd}	150	70	ns
I_{cc}	0,5	1,0	mA/CI

Fig. 4.25

Contém quatro portas AND de duas entradas.

4.4 - Análise de Circuitos Combinacionais

A análise de um circuito consiste em determinar a relação entre sua(s) saída(s) e suas entradas. De forma objetiva, analisar um circuito consiste em determinar o que ele faz.

Ao analisarmos um circuito combinacional, procuramos obter a expressão lógica ou a tabela da verdade para sua saída ou saídas. Neste item, iremos analisar diversos circuitos combinacionais, ilustrando através de exemplos a sistemática de análise.

Exemplo 4.4.1

Determinar a equação da saída Y para o circuito da figura 4.26.

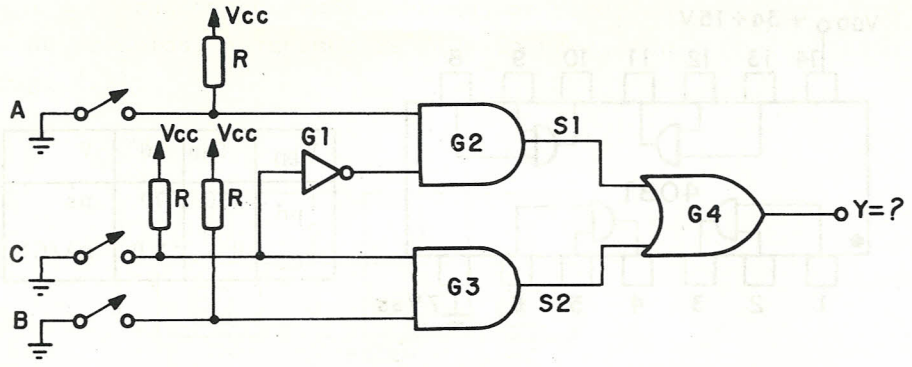


Fig. 4.26 - Circuito do exemplo 4.4.1

Solução:

A saída de G_2 será:

$$S_1 = A \cdot \bar{C}$$

A saída de G_3 será:

$$S_2 = B \cdot C$$

como

$$Y = S_1 + S_2$$

tem-se

$$Y = A \cdot \bar{C} + B \cdot C$$

Fazendo a tabela da verdade para este circuito obtemos:

C	B	A	S_1	S_2	Y
Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	1	Ø	1
Ø	1	Ø	Ø	Ø	Ø
Ø	1	1	1	Ø	1
1	Ø	Ø	Ø	Ø	Ø
1	Ø	1	Ø	Ø	Ø
1	1	Ø	Ø	1	1
1	1	1	Ø	1	1

Fig. 4.27 - Tabela da Verdade para o Circuito da fig. 4.26.

Observando a tabela, verificamos que a saída Y é igual à entrada A, para C=Ø e igual à entrada B, para C=1. Em outras palavras, de acordo com o estado da entrada C, podemos selecionar uma das duas entradas, A ou B, e colocá-la na saída. Um circuito deste tipo encontra grande aplicação na prática e recebe o nome de multiplexor ou DATA SELECTOR, ou ainda multiplexador.

Trata-se, neste caso, de um multiplexor de duas entradas para uma saída (2-to-1 MUX). A entrada C é a linha de controle, que permite transferir para a saída ou a entrada A (C=Ø) ou a entrada B (C=1).

Exemplo 4.4.2

Determinar as equações das saídas Y_0, Y_1, Y_2 e Y_3 para o circuito da fig. 4.28.

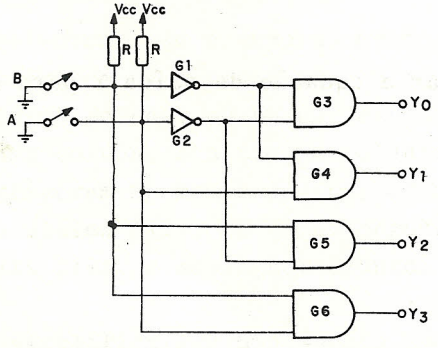


Fig. 4.28 - Circuito do exemplo 4.4.2.

Solução:

Inspecionando o circuito da fig. 4.28, tem-se imediatamente:

$$Y_0 = \bar{B} \cdot \bar{A}$$
$$Y_1 = \bar{B} \cdot A$$
$$Y_2 = B \cdot \bar{A}$$
$$Y_3 = B \cdot A$$

e a tabela da verdade é mostrada na fig. 4.29.

B	A	Y_0	Y_1	Y_2	Y_3
Ø	Ø	1	Ø	Ø	Ø
Ø	1	Ø	1	Ø	Ø
1	Ø	Ø	Ø	1	Ø
1	1	Ø	Ø	Ø	1

Fig. 4.29 - Tabela da Verdade para o Circuito da fig. 4.28.

A tabela nos mostra que somente uma das saídas estará ativa (igual a "1") em determinado instante e esta saída corresponde ao valor binário representado pelas entradas B e A. Este é outro circuito que encontra grande aplicação prática, sendo denominado decodificador ou DECODER, neste caso, um decodificador de duas linhas para quatro linhas (2-to-4 DECODER).

Exemplo 4.4.3

Determinar a equação da saída Y para o circuito da fig. 4.30.

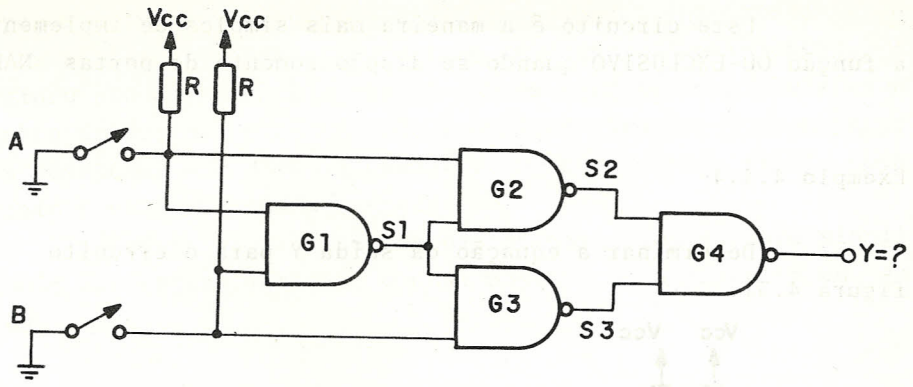


Fig. 4.30 - Circuito do exemplo 4.4.3.

Solução:

Examinando o circuito, temos:

$$S_1 = \overline{A \cdot B} = \bar{A} + \bar{B}$$
$$S_2 = \overline{A \cdot S_1}$$
$$S_3 = \overline{B \cdot S_1}$$

e para a saída:

$$Y = \overline{S_2 \cdot S_3} = \overline{S_2} + \overline{S_3} = (\overline{\overline{A \cdot S_1}}) + (\overline{\overline{B \cdot S_1}})$$

$$= A \cdot S_1 + B \cdot S_1 = (A + B) \cdot S_1$$

$$= (A + B) \cdot (\bar{A} + \bar{B}) =$$

$$= \underbrace{A \cdot \bar{A}}_{\emptyset} + A \cdot \bar{B} + B \cdot \bar{A} + \underbrace{B \cdot \bar{B}}_{\emptyset}$$

$$\Rightarrow \quad Y = A \cdot \bar{B} + \bar{A} \cdot B$$

ou seja, o circuito realiza a função OU-EXCLUSIVO entre A e B :

$$Y = A \oplus B$$

Este circuito é a maneira mais simples de implementar a função OU-EXCLUSIVO quando se dispõe somente de portas NAND.

Exemplo 4.4.4

Determinar a equação da saída Y para o circuito da figura 4.31.

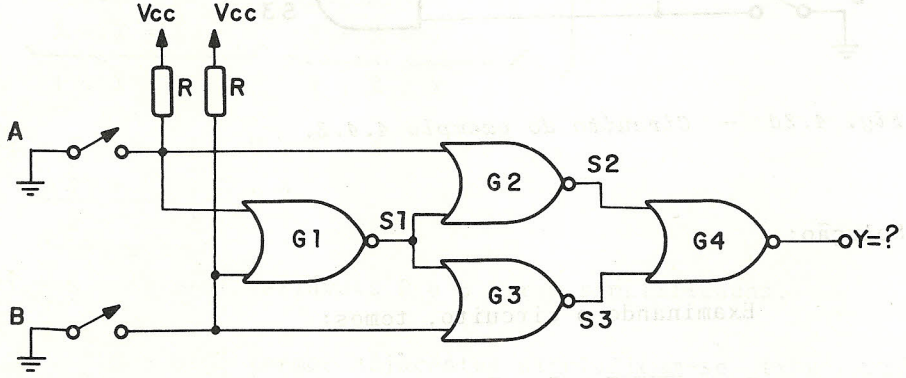


Fig. 4.31 - Circuito do exemplo 4.4.4.

Solução:

Examinando o circuito, temos:

$$S_1 = \overline{A + B} = \bar{A} \cdot \bar{B}$$
$$S_2 = \overline{A + S_1}$$
$$S_3 = \overline{B + S_1}$$

e, para a saída

$$Y = \overline{S_2 + S_3} = \overline{S_2} \cdot \overline{S_3}$$
$$= \overline{(\overline{A + S_1})} \cdot \overline{(\overline{B + S_1})} = (A + S_1) \cdot (B + S_1)$$
$$= A \cdot B + A \cdot S_1 + B \cdot S_1 + S_1 \cdot S_1$$
$$= A \cdot B + S_1 \cdot \underbrace{(A + B + 1)}_{= 1}$$
$$= A \cdot B + S_1$$

$$\Rightarrow \quad Y = A \cdot B + \bar{A} \cdot \bar{B}$$

ou seja, o circuito realiza a função de equivalência ou comparação entre A e B.

$$Y = A \odot B$$

Este circuito é a maneira mais simples de efetuar a função coincidência quando se dispõe somente de portas NOR.

Os exemplos 4.4.3 e 4.4.4 ilustram a análise de circuitos combinacionais através da manipulação de expressões lógicas. Este método torna-se muito trabalhoso para circuitos mais complexos, nos quais as expressões resultantes tornam-se muito extensas. Neste caso, o levantamento da tabela da verdade é o método mais indicado.

Exemplo 4.4.5

Determinar a tabela da verdade para o circuito da fig. 4.32.

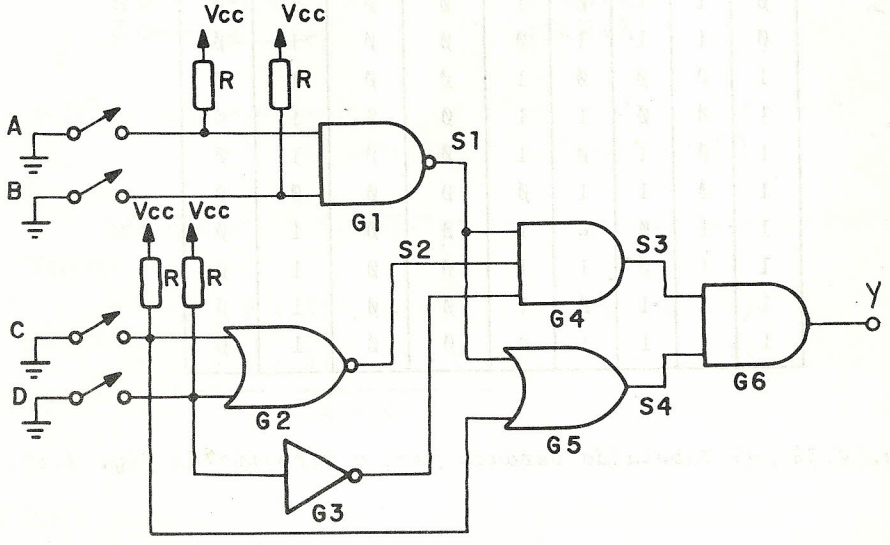


Fig. 4.32 - Circuito do exemplo 4.4.5.

Solução:

Da observação do circuito, temos:

$S_1 = \overline{A \cdot B}$

$S_2 = \overline{C + D}$

$S_3 = S_1 \cdot S_2 \cdot \bar{D}$

$S_4 = S_1 + C$

$Y = S_3 \cdot S_4$

Montando a tabela da verdade para o circuito, incluindo os termos parciais e a saída final, obtemos a fig. 4.33:

D	C	B	A	S_1	S_2	S_3	S_4	Y
Ø	Ø	Ø	Ø	1	1	1	1	1
Ø	Ø	Ø	1	1	1	1	1	1
Ø	Ø	1	Ø	1	1	1	1	1
Ø	Ø	1	1	Ø	1	Ø	Ø	Ø
Ø	1	Ø	Ø	1	Ø	Ø	1	Ø
Ø	1	Ø	1	1	Ø	Ø	1	Ø
Ø	1	1	Ø	1	Ø	Ø	1	Ø
Ø	1	1	1	Ø	Ø	Ø	1	Ø
1	Ø	Ø	Ø	1	Ø	Ø	1	Ø
1	Ø	Ø	1	1	Ø	Ø	1	Ø
1	Ø	1	Ø	1	Ø	Ø	1	Ø
1	Ø	1	1	Ø	Ø	Ø	Ø	Ø
1	1	Ø	Ø	1	Ø	Ø	1	Ø
1	1	Ø	1	1	Ø	Ø	1	Ø
1	1	1	Ø	1	Ø	Ø	1	Ø
1	1	1	1	Ø	Ø	Ø	1	Ø

Fig. 4.33 - Tabela da verdade para o Circuito da fig. 4.32.

4.5 - Projeto de Circuitos Combinacionais

4.5.1 - Introdução

O projeto de um circuito combinacional constitui-se num problema de síntese, em oposição ao de análise estudado no item anterior. O problema de síntese consiste em determinar um circuito que realize uma função desejada. De modo geral, podemos distinguir as seguintes fases no projeto de um circuito combinacional:

a) definição do problema - estabelecemos aqui qual a função que se deseja executar, definimos as entradas e saídas envolvidas;

b) colocação do problema em termos de uma tabela da verdade ou de equações lógicas que forneçam a saída ou saídas desejadas em função das entradas disponíveis;

c) obtenção do circuito a partir da tabela da verdade ou das equações encontradas no passo anterior, procurando-se minimizar este circuito tanto quanto possível;

d) implementação prática do circuito, levando em conta fatores como consumo de potência, velocidade requerida, componentes disponíveis, etc.

A seqüência acima descrita é bastante genérica e num projeto real devem ser examinadas as diversas alternativas possíveis, cabendo ao projetista determinar o circuito que atende às exigências do projeto, com o menor custo, dentro da situação específica que se apresenta.

Procuraremos ilustrar através de diversos exemplos as técnicas de projeto de circuitos combinacionais. Neste capítulo, limitaremos a abordagem à implementação de circuitos combinacionais com portas lógicas e, em capítulos posteriores, mostraremos outras técnicas mais poderosas.

4.5.2 - Obtenção de Expressões Lógicas a partir da Tabela da Verdade.

Uma vez definido o problema, em termos da saída ou

saídas desejadas em função das entradas, temos montada uma tabela da verdade para o circuito que se quer projetar. Partindo da tabela, podemos obter as expressões lógicas para as saídas. Neste item, examinaremos duas possibilidades:

a) expressar a função lógica como uma operação "OU" de diversos termos, cada um dos quais é uma expressão "E" das variáveis de entrada (forma da soma de produtos);
b) expressar a função lógica como uma operação "E" de diversos termos, cada um dos quais é uma operação "OU" das variáveis de entrada (forma do produto de somas).

Vamos ilustrar os dois métodos através de exemplos.

Exemplo 4.5.1

Dada a tabela da verdade da fig. 4.34, determinar a função lógica correspondente à saída Y expressando-a na forma de uma soma de produtos.

	C	B	A	Y
linha Ø →	Ø	Ø	Ø	Ø
linha 1 →	Ø	Ø	1	1
linha 2 →	Ø	1	Ø	1
linha 3 →	Ø	1	1	Ø
linha 4 →	1	Ø	Ø	1
linha 5 →	1	Ø	1	Ø
linha 6 →	1	1	Ø	Ø
linha 7 →	1	1	1	1

Fig. 4.34 - Tabela da Verdade para o exemplo 4.5.1.

Solução:

A saída Y será igual a 1 para as combinações das entradas indicadas nas linhas 1, 2, 4 e 7 da tabela, ou seja:

Y = 1 para

combinação das entradas expressa na linha 1

"OU"

combinação das entradas expressa na linha 2

"OU"

combinação das entradas expressa na linha 4

"OU"

combinação das entradas expressa na linha 7

Podemos, assim, encarar a tabela da fig. 4.34 como a operação "OU" de quatro funções cujas tabelas da verdade teriam apenas a linha 1, apenas a linha 2, apenas a linha 4 e apenas a linha 7 iguais a 1, respectivamente. Isto está ilustrado na fig. 4.35:

C	B	A	Y
Ø	Ø	Ø	Ø
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	Ø
1	Ø	Ø	1
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	1

=

C	B	A	S_1
Ø	Ø	Ø	Ø
Ø	Ø	1	1
Ø	1	Ø	Ø
Ø	1	1	Ø
1	Ø	Ø	Ø
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	Ø

+

C	B	A	S_2
Ø	Ø	Ø	Ø
Ø	Ø	1	Ø
Ø	1	Ø	1
Ø	1	1	Ø
1	Ø	Ø	Ø
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	Ø

+

C	B	A	S_2
Ø	Ø	Ø	Ø
Ø	Ø	1	Ø
Ø	1	Ø	Ø
Ø	1	1	Ø
1	Ø	Ø	1
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	Ø

+

C	B	A	S_4
Ø	Ø	Ø	Ø
Ø	Ø	1	Ø
Ø	1	Ø	Ø
Ø	1	1	Ø
1	Ø	Ø	Ø
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	1

Da fig. 4.35, concluímos imediatamente que:

$S_1 = \overline{C} \cdot \overline{B} \cdot A$

$S_2 = \overline{C} \cdot B \cdot \overline{A}$

$S_3 = C \cdot \overline{B} \cdot \overline{A}$

$S_4 = C \cdot B \cdot A$

e, sendo

$$Y = S_1 + S_2 + S_3 + S_4$$

vem

$$Y = \overline{C} \cdot \overline{B} \cdot A + \overline{C} \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A} + C \cdot B \cdot A$$

Obtivemos, assim, a função lógica representada na tabela da figura 4.34, expressa como uma soma de produtos.

Observe que:

$S_1 = \overline{C} \cdot \overline{B} \cdot A$ ← corresponde à combinação C = Ø, B = Ø, A = 1

$S_2 = \overline{C} \cdot B \cdot \overline{A}$ ← corresponde à combinação C = Ø, B = 1, A = Ø

$S_3 = C \cdot \overline{B} \cdot \overline{A}$ ← corresponde à combinação C = 1, B = Ø, A = Ø

$S_4 = C \cdot B \cdot A$ ← corresponde à combinação C = 1, B = 1, A = 1

Todo o processo ilustrado acima nos permite deduzir um método prático para obtermos uma função lógica expressa como soma de produtos.

1) Verificamos as combinações das entradas para as quais a função é igual a 1;

2) expressamos cada uma das combinações acima como uma operação "E" das variáveis de entrada, onde aquelas iguais a "Ø" aparecem barradas;

3) expressamos a função lógica como uma operação "OU" dos termos obtidos no passo 2.

A expressão de uma função lógica como uma soma de produtos é a mais indicada quando se vai implementá-la utilizando portas NAND.

Exemplo 4.5.2

Expressar a função lógica do exemplo anterior como um produto de somas.

Solução:

A saída Y será igual a "Ø" para as combinações das entradas indicadas nas linhas Ø, 3, 5 e 6 da tabela da fig.4.34, ou seja:

Y = Ø para

combinação das entradas expressa na linha Ø

"E"

combinação das entradas expressa na linha 3

"E"

combinação das entradas expressa na linha 5

"E"

combinação das entradas expressa na linha 6

Podemos assim encarar a tabela da fig. 4.34 como a operação "E" de quatro funções cujas tabelas da verdade teriam apenas a linha "Ø", apenas a linha 3, apenas a linha 5 e apenas a linha 6 iguais a "Ø", respectivamente. Isto está ilustrado na fig. 4.36:

C	B	A	Y
Ø	Ø	Ø	Ø
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	Ø
1	Ø	Ø	1
1	Ø	1	Ø
1	1	Ø	Ø
1	1	1	1

=

C	B	A	P_1
Ø	Ø	Ø	Ø
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	1
1	Ø	Ø	1
1	Ø	1	1
1	1	Ø	1
1	1	1	1

·

C	B	A	P_2
Ø	Ø	Ø	1
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	Ø
1	Ø	Ø	1
1	Ø	1	1
1	1	Ø	1
1	1	1	1

C	B	A	P_3
Ø	Ø	Ø	1
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	1
1	Ø	Ø	1
1	Ø	1	Ø
1	1	Ø	1
1	1	1	1

C	B	A	P_4
Ø	Ø	Ø	1
Ø	Ø	1	1
Ø	1	Ø	1
Ø	1	1	1
1	Ø	Ø	1
1	Ø	1	1
1	1	Ø	Ø
1	1	1	1

Fig. 4.36 - Tabela da fig. 4.34 como resultante da operação "E" de quatro funções.

Da fig. 4.36 concluímos imediatamente que

$$P_1 = C + B + A$$
$$P_2 = C + \bar{B} + \bar{A}$$
$$P_3 = \bar{C} + B + \bar{A}$$
$$P_4 = \bar{C} + \bar{B} + A$$

de onde se conclui que

$$Y = P_1 \cdot P_2 \cdot P_3 \cdot P_4$$

e

$$Y = (C+B+A) \cdot (C+\bar{B}+\bar{A}) \cdot (\bar{C}+B+\bar{A}) \cdot (\bar{C}+\bar{B}+A)$$

Obtivemos assim a função lógica representada na tabela da fig. 4.34 expressa como um produto de somas.

Observe que

$P_1 = C + B + A$ ← corresponde à combinação C=Ø, B=Ø, A=Ø
$P_2 = C + \bar{B} + \bar{A}$ ← corresponde à combinação C=Ø, B=1, A=1
$P_3 = \bar{C} + B + \bar{A}$ ← corresponde à combinação C=1, B=Ø, A=1
$P_4 = \bar{C} + \bar{B} + A$ ← corresponde à combinação C=1, B=1, A=Ø

Todo o processo ilustrado acima nos permite deduzir um método prático para obtermos uma função lógica expressa como um produto de somas:

1) verificamos as combinações das entradas para as quais a função é igual a "Ø";

2) expressamos cada uma das combinações acima como uma operação "OU" das variáveis de entrada, onde aquelas iguais a "1" aparecem barradas;

3) expressamos a função lógica como uma operação "E" dos termos obtidos no passo 2.

A expressão de uma função lógica como um produto de somas é a mais indicada quando se vai implementá-la utilizando portas NOR.

4.5.3 - Implementação de Circuitos Combinacionais com portas NAND e portas NOR.

Estudamos, no capítulo 2, a utilização de portas NAND e NOR como bloco lógico universal, mostrando que qualquer outra função lógica pode ser conseguida a partir delas.

Quando se obtém a expressão lógica de uma saída a partir da tabela da verdade, expressa na forma de uma soma de produtos, a configuração do circuito com portas NAND é encontrada de imediato.

Por exemplo, consideremos a expressão:

$$Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A$$

a qual se deseja implementar com portas NAND. Cada produto, na expressão acima, pode ser obtido com uma porta NAND com a saída invertida, que equivale a uma porta "E". A soma (operação "OU") é feita aplicando as entradas invertidas a uma porta NAND. A fig. 4.37 ilustra este raciocínio:

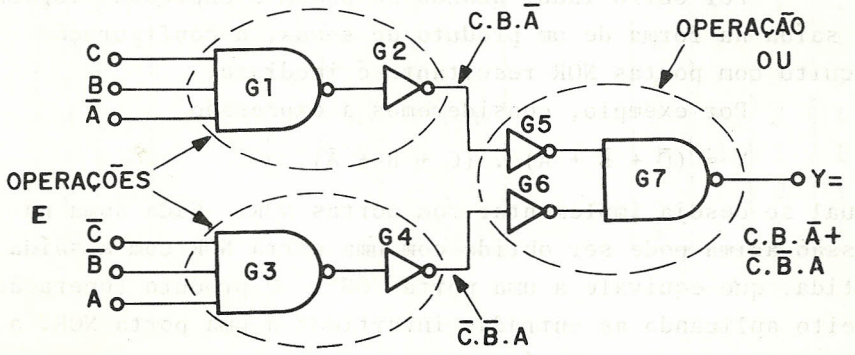


Fig. 4.37 - Implementação direta de $Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A$

Observe que a dupla inversão resulta na variável original, de modo que o circuito da fig. 4.37 pode ser simplificado para o da fig. 4.38:

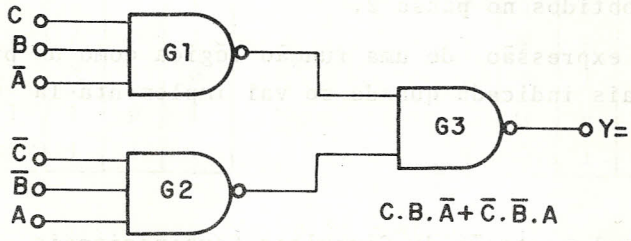


Fig. 4.38 - Implementação Prática de $Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A$

Deste modo, a inversão necessária na saída de uma porta NAND (para realizar a operação "E") é seguida de outra na entrada da porta NAND seguinte (para realizar a operação "OU")e ambas se anulam. Quando se tem a função lógica expressa na forma de uma soma de produtos e se deseja construir com portas NAND um circuito que a realize é necessário tão somente:

1) aplicar cada uma das combinações das entradas que correspondem aos produtos a uma porta NAND;

2) utilizar as saídas das portas NAND obtidas acima como entradas para uma porta NAND cuja saída será a função desejada.

Requer-se, pois, apenas dois níveis de lógica para implementar a função.

Por outro lado, quando se obtém a expressão lógica de uma saída na forma de um produto de somas, a configuração do circuito com portas NOR resultante é imediata.

Por exemplo, consideremos a expressão

$$Y = (\bar{C} + \bar{B} + A) \cdot (C + B + \bar{A})$$

a qual se deseja implementar com portas NOR. Cada soma na expressão acima pode ser obtida com uma porta NOR com a saída invertida, que equivale a uma porta "OU". O produto (operação"E") é feito aplicando as entradas invertidas a uma porta NOR. a fig 4.39 ilustra este raciocínio:

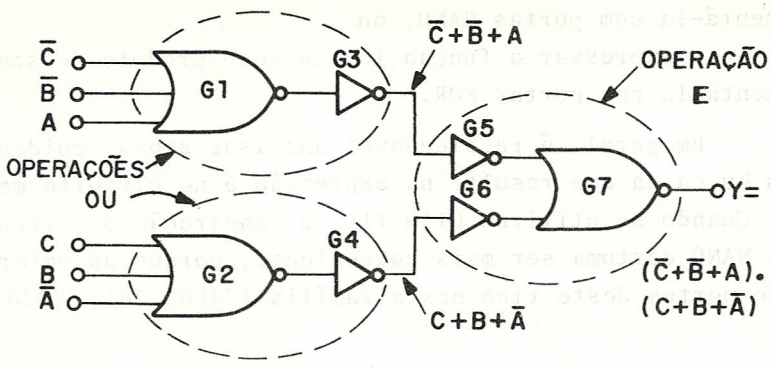


Fig. 4.39 - Implementação Direta de Y = (C̄+B̄+A) . (C+B+Ā).

Como a dupla inversão resulta na variável original, o circuito da fig. 4.39 pode ser simplificador para o da fig.4.40:

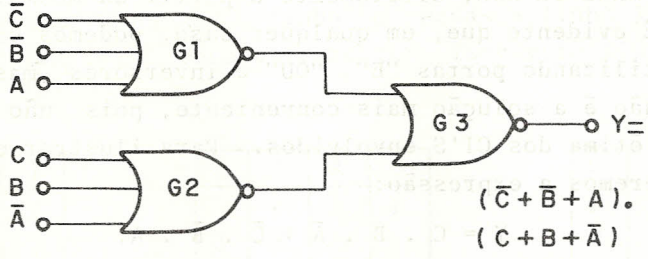


Fig. 4.40 - Implementação Prática de Y = (C̄+B̄+A) . (C+B+Ā).

De maneira prática, para implementar com portas NOR uma função lógica expressa na forma de produto de somas basta:

1) aplicar cada uma das combinações das entradas que corresponde a uma soma a uma porta NOR;

2) utilizar as saídas das portas NOR obtidas acima como entradas para uma porta NOR cuja saída será a função desejada.

Requerem-se apenas dois níveis de lógica para implementar a função.

Neste ponto, é natural que se pergunte qual das duas alternativas apresentadas é mais vantajosa:

- expressar a função lógica como soma de produtos e

implementá-la com portas NAND, ou

- expressar a função lógica como produto de somas e implementá-la com portas NOR.

Em geral, é recomendável analisar ambas cuidadosamente, em busca da que resulte na expressão e no circuito mais simples. Quando se utilizam CI's TTL, a construção do circuito com portas NAND costuma ser mais conveniente, porque há maior variedade de portas deste tipo nesta família (7400, 7410, 7420, 7430, etc).

4.5.4 - Simplificação e Minimização

Uma vez obtida a expressão lógica de uma saída a partir da tabela da verdade, já vimos como construir o circuito com portas NAND ou NOR, diretamente a partir da mesma.

É evidente que, em qualquer caso, podemos construir o circuito utilizando portas "E", "OU" e inversores ,mas, em geral, esta não é a solução mais conveniente, pois não permite a utilização ótima dos CI'S envolvidos. Para ilustrar este ponto, consideremos a expressão:

$$Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A.$$

Utilizando portas "E" e "OU" para construir um circuito que realiza esta função, obtemos o esquema ilustrado na fig. 4.41:

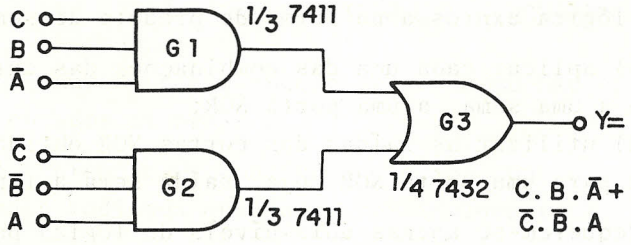


Fig. 4.41 - $Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A$ implementada com portas "E" e "OU".

Na fig. 4.41, admitiu-se como exemplo o uso de um CI 7411 (TRIPLE 3-INPUT AND GATE) e de um CI 7432 (QUAD 2-INPUT OR GATE). Foram usados dois CI'S, ficando uma porta "E" no 7411 e três portas "OU" no 7432 sem utilização.

A mesma função implementada somente com portas NAND requereria um único CI 7410 (TRIPLE 3-INPUT NAND GATE), ligado conforme mostra a fig. 4.42:

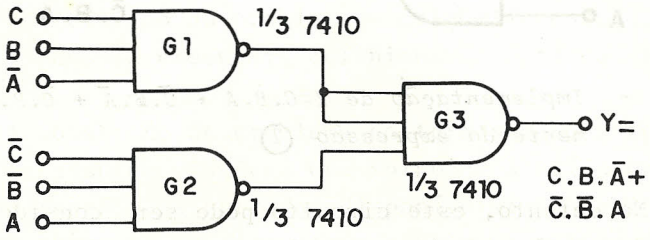


Fig. 4.42 - $Y = C \cdot B \cdot \bar{A} + \bar{C} \cdot \bar{B} \cdot A$ Implementada com portas NAND.

E claro que o circuito da fig. 4.42 é mais conveniente que o da fig. 4.41, pois representa menor custo e menor consumo.

Por este motivo, no projeto de um circuito puramente combinacional procura-se, na maioria dos casos, construir o mesmo exclusivamente com portas NAND ou NOR.

O circuito que se pode derivar diretamente da expressão lógica de uma saída, quando a mesma está na forma de uma soma de produtos ou produto de somas não é, entretanto, necessariamente o mais simples possível.

Exemplo 4.5.4.1:

Consideremos a expressão:

$$Y = C \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A} + \bar{C} \cdot B \cdot A \qquad (1)$$

Aplicando o método apresentado no item 4.5.3, resulta o circuito da fig. 4.43:

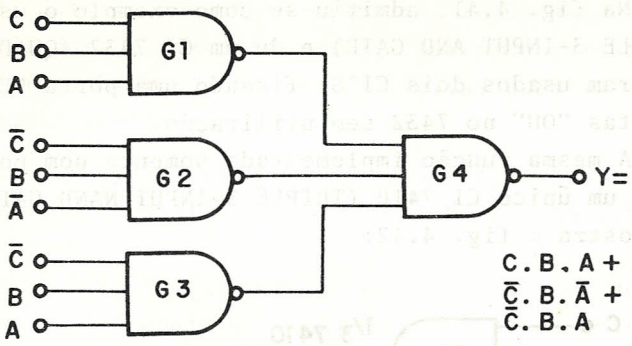


Fig. 4.43 - Implementação de Y=C.B.A + C̄.B.Ā + C̄.B.A diretamente da expressão (1)

No entanto, este circuito pode ser consideravelmente simplificado se observarmos que alguns termos da expressão 1 podem ser agrupados. Assim:

$$Y = C \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A} + \bar{C} \cdot B \cdot A$$

$$= C \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A} + \underbrace{\bar{C} \cdot B \cdot A + \bar{C} \cdot B \cdot A}_{X + X = X}$$

$$Y = C \cdot B \cdot A + \bar{C} \cdot B \cdot A + \bar{C} \cdot B \cdot \bar{A} + \bar{C} \cdot B \cdot A$$

$$= \underbrace{(C+\bar{C})}_{X + \bar{X} = 1} \cdot B \cdot A + \bar{C} \cdot B \cdot \underbrace{(A+\bar{A})}_{X + \bar{X} = 1}$$

$$1 \cdot X = X$$

$$Y = B \cdot A + \bar{C} \cdot B,$$

resultando no circuito da fig. 4.44:

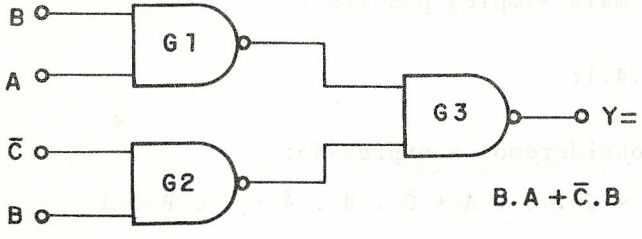


Fig. 4.44 - Circuito Resultante da Simplificação de Y = C . B . A + C̄ . B . Ā + C̄ . B . A.

Os circuitos das figs 4.43 e 4.44 realizam exatamente a mesma função, sendo o segundo mais simples e, portanto, de menor custo.

Exemplo 4.5.4.2

Vejamos um outro exemplo: Seja a expressão

$$Y = (\bar{C} + B + \bar{A}).(C + B + \bar{A}).(\bar{C} + \bar{B} + \bar{A}) \qquad (2)$$

da qual resulta, aplicando o método do item 4.5.3, o circuito da fig. 4.45.

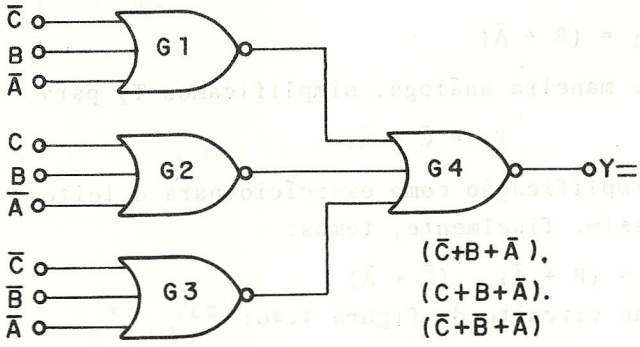


Fig. 4.45 - Implementação de $Y = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$ *diretamente da expressão* (2)

Neste caso, a simplificação é um pouco mais trabalhosa. Temos:

$$Y = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$$

$$= \underbrace{(\bar{C} + B + \bar{A}) \cdot (\bar{C} + B + \bar{A})}_{X \,.\, X = X} \cdot (C + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$$

$$Y = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A}) \cdot (\bar{C} + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$$

façamos

$$T_1 = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A})$$

$$T_2 = (\bar{C} + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$$

e simplifiquemos primeiramente T_1

$$T_1 = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A})$$

$$= \underbrace{\bar{C} \cdot C}_{X \cdot \bar{X} = \emptyset} + \bar{C}.(B+\bar{A}) + (B+\bar{A}).C + \underbrace{(B+\bar{A}) \cdot (B+\bar{A})}_{X \cdot X = X}$$

$$\emptyset + X = X$$

$$\Rightarrow \quad T_1 = \bar{C}.(B + \bar{A}) + (B + \bar{A}).C + (B + \bar{A})$$

$$= (B + \bar{A}) \cdot \underbrace{(\bar{C} + C)}_{X + \bar{X} = 1} + (B + \bar{A})$$

$$X \cdot 1 = X$$

$$\Rightarrow \quad T_1 = \underbrace{(B + \bar{A}) + (B + \bar{A})}_{X + X = X}$$

$$\Rightarrow \quad T_1 = (B + \bar{A})$$

De maneira análoga, simplificamos T_2 para

$$T_2 = \bar{C} + \bar{A},$$

ficando a simplificação como exercício para o leitor.

Assim, finalmente, temos:

$$Y = (B + \bar{A}) \cdot (\bar{C} + \bar{A}),$$

resultando no circuito da figura 4.46:

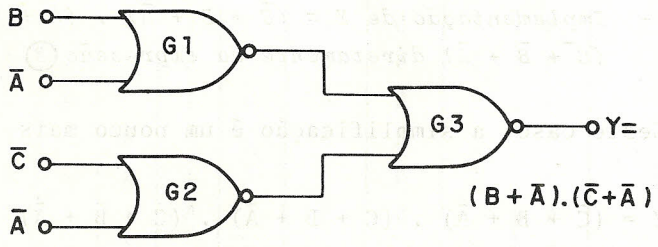


Fig. 4.46 - Circuito Resultante da Simplificação de $Y = (\bar{C} + B + \bar{A}) \cdot (C + B + \bar{A}) \cdot (\bar{C} + \bar{B} + \bar{A})$.

O circuito da figura 4.46 realiza exatamente a mesma função que o da fig. 4.45 e é consideravelmente mais simples, sendo, portanto, preferível.

Os dois exemplos anteriores ilustram a simplificação de funções lógicas através de manipulações algébricas, Como se

vê pelos exemplos, este processo é trabalhoso, envolvendo o uso de artifícios, como a repetição de termos já existentes, e exigindo uma certa familiaridade com os teoremas e postulados da álgebra de Boole. O método não é prático para funções um pouco mais complexas e é difícil garantir que a expressão final seja de fato a mais simples possível.

Felizmente, existe um método mais simples para simplificação de funções lógicas, o qual passaremos a discutir em seguida.

4.5.5 - Mapas de Karnaugh

Nos dois exemplos do item 4.5.4, podemos observar que a simplificação resulta de haver termos que diferem entre si por apenas uma variável. Dois termos que se enquadrem nesta categoria são chamados adjacentes e resultam na simplificação da variável pela qual diferem. Por exemplo:

$$Y = C \cdot B \cdot A + \bar{C} \cdot B \cdot A$$
$$= B \cdot A \cdot \underbrace{(C + \bar{C})}_{X + \bar{X} = 1}$$

} os dois termos diferem pela variável C

$\Rightarrow$ $Y = B \cdot A$ ⟶ a variável C foi simplificada

Um outro exemplo:

$$Y = (\bar{C} + B + A).(\bar{C} + B + \bar{A})$$

os dois termos diferem pela variável A

$$= \underbrace{(\bar{C} + B).(\bar{C} + B)}_{X \cdot X = X} + (\bar{C} + B).\bar{A} + A.(\bar{C} + B) + \underbrace{A \cdot \bar{A}}_{X \cdot \bar{X} = \emptyset}$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{X + \emptyset = X}$$

$$\Rightarrow \quad Y = (\bar{C} + B) + (\bar{C} + B) \cdot \underbrace{(A + \bar{A})}_{X + \bar{X} = 1}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{X \cdot 1 = X}$$

$$\Rightarrow \quad Y = \underbrace{(\bar{C} + B) \cdot (\bar{C} + B)}_{X \cdot X = X}$$

$\Longrightarrow$ $Y = (\bar{C} + B)$ $\longrightarrow$ a variável A foi simplificada

Havendo quatro termos adjacentes, é possível a simplificação de duas variáveis, aquelas que são diferentes entre os quatro termos. Por exemplo:

$$Y = D.C.B.A + \bar{D}.C.B.A + \bar{D}.C.\bar{B}.A + D.C.\bar{B}.A$$

$$= \underbrace{(D + \bar{D})}_{X + \bar{X} = 1} .C.B.A + \underbrace{(D + \bar{D})}_{X + \bar{X} = 1}. C.\bar{B}.A$$

$$1 \;.\; X = X \qquad\qquad 1 \;.\; X = X$$

os quatro termos diferem pelas variáveis D e B.

$\Longrightarrow Y = C \;.\; B \;.\; A + C \;.\; \bar{B} \;.\; A$

$$= \underbrace{(B + \bar{B})}_{X + \bar{X} = 1} \;.\; C \;.\; A$$

$$1 \;.\; X = X$$

$\Longrightarrow Y = C \;.\; A$ $\longrightarrow$ as variáveis D e B foram simplificadas.

Com oito termos adjacentes simplificam-se três variáveis, com dezesseis simplificam-se quatro variáveis e, em geral, com "2^n" termos adjacentes simplificam-se "n" variáveis.

Os mapas de Karnaugh são diagramas nos quais as adjacências entre os termos são facilmente identificáveis. A tabela da verdade é plotada sobre o mapa, permitindo a identificação dos termos adjacentes e a simplificação da expressão lógica.

A fig. 4.47 mostra os mapas de Karnaugh para 2,3,4 e 5 variáveis, com as adjacências indicadas pelas setas:

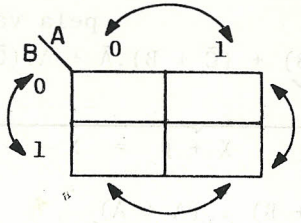


Fig. 4.47 - (a) Mapa da Karnaugh para 2 variáveis.

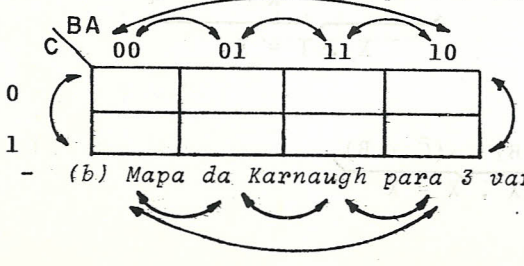


Fig. 4.47 - (b) Mapa da Karnaugh para 3 variáveis.

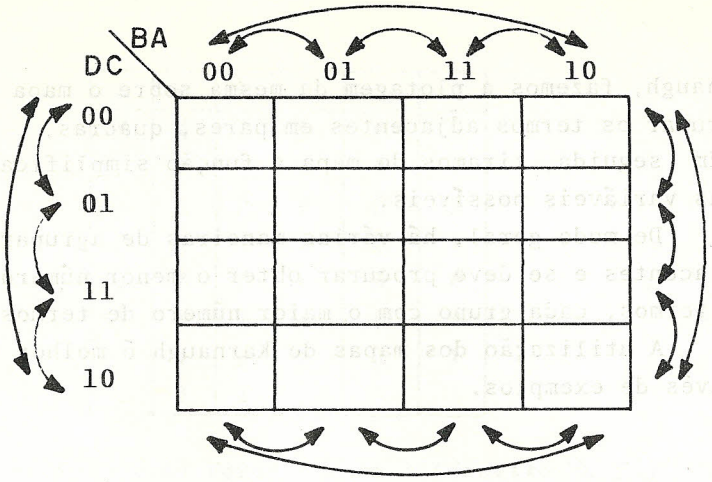


Fig. 4.47 - (c) Mapa de Karnaugh para 4 variáveis.

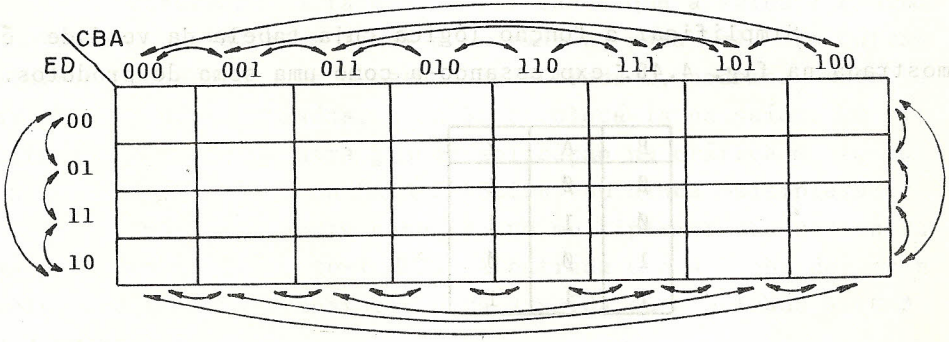


Fig. 4.47 - (d) Mapa de Karnaugh para 5 variáveis.

Fig. 4.47 - Mapas de Karnaugh para 2,3,4 e 5 variáveis.

Nos mapas de Karnaugh mostrados na fig. 4.47, vemos que dois termos são adjacentes quando:

a) estiverem lado a lado na mesma fila ou coluna;
b) estiverem na mesma fila, um na primeira e outro na última coluna, ou um na segunda e outro na penúltima coluna, e assim por diante;
c) estiverem na mesma coluna, um na primeira e outro na última fila, ou um na segunda e outro na penúltima fila, e assim por diante.

Para simplificarmos uma função lógica através do mapa

de Karnaugh, fazemos a plotagem da mesma sobre o mapa e procuramos agrupar os termos adjacentes em pares, quadras, óctuplos, etc. Em seguida, tiramos do mapa a função simplificada, eliminando as variáveis possíveis.

De modo geral, há várias maneiras de agrupar os termos adjacentes e se deve procurar obter o menor número de grupos de termos, cada grupo com o maior número de termos possível.

A utilização dos mapas de Karnaugh é melhor esclarecida através de exemplos.

Exemplo 4.5.5.1:

Simplificar a função lógica cuja tabela da verdade é mostrada na fig. 4.48, expressando-a como uma soma de produtos.

B	A	Y
Ø	Ø	1
Ø	1	1
1	Ø	Ø
1	1	1

Fig. 4.48 - Tabela da Verdade para a função do exemplo 4.5.5.1

Solução:

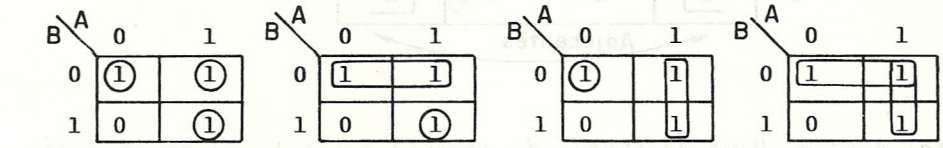
Construindo o mapa de Karnaugh para duas variáveis e plotando a tabela sobre ele, obteremos a fig. 4.49:

B \ A	0	1
0	1	1
1	0	1

Fig. 4.49 - Tabela da Verdade da fig. 4.48 plotada sobre o mapa.

Como desejamos obter a função simplificada expressa na forma de uma soma de produtos, devemos agrupar os termos correspondentes ao valor 1. Os termos adjacentes podem ser agrupados nas formas mostradas na fig. 4.50, onde se indicam também as expressões resultantes em cada caso.

Estas expressões são encontradas aplicando as regras explicadas no item 4.5.2 e considerando as simplificações decorrentes das adjacências de termos.



Y = $\overline{B}.\overline{A}+\overline{B}.A+B.A$ (a)

Y = $\overline{B}+B.A$ (b)

Y = $\overline{B}.\overline{A}+A$ (c)

Y = $\overline{B}+A$ (d)

Fig. 4.50 - Grupamento de Termos no Mapa.

É evidente que a forma (d) permite a maior simplificação, pois conseguimos duas duplas; repare que a forma (a) representa a função não simplificada, como seria obtida diretamente da tabela.

Exemplo 4.5.5.2:

Simplificar a função

$Y = C \cdot B \cdot \overline{A} + C \cdot \overline{B} \cdot \overline{A}$

pelo mapa de Karnaugh, expressando-a
1º) como soma de produtos,
2º) como produto de somas.

Solução:

Observer que os produtos $C.B.\overline{A}$ e $C.\overline{B}.\overline{A}$ correspondem às combinações 110 e 100 respectivamente. Construindo o mapa e plotando a função sobre ele, obtemos a fig. 4.51.

Para conseguirmos a expressão final na forma de uma soma de produtos, agrupamos os "1"s do mapa, conforme ilustrado na fig. 4.51:

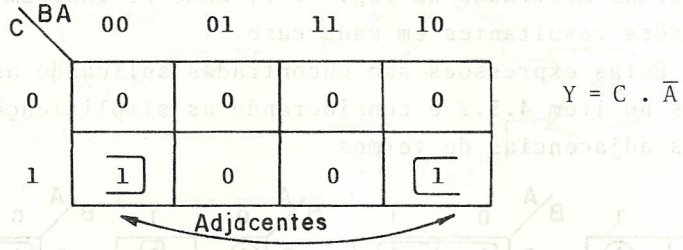


Fig. 4.51 - Mapa da Função do exemplo 4.5.5.2 e simplificação pelos "1"s.

Para conseguirmos a expressão final na forma de um produto de somas, agrupamos os termos iguais a Ø. Este agrupamento pode ser feito de várias formas, mas a que resulta na maior simplificação é aquela ilustrada na fig. 4.52:

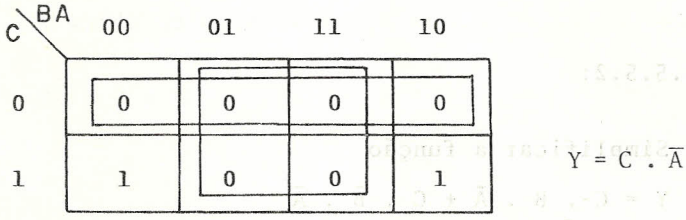


Fig. 4.52 - Mapa da Função do exemplo 4.5.5.2 e simplificação pelos "Ø"s.

Neste exemplo, as expressões na forma de soma de produtos e produto de somas são exatamente iguais. Observe que, ao fazermos a simplificação pelos "1"s, agrupamos dois produtos de três variáveis e simplificamos a variável B, resultando num único produto C.Ā; já ao fazermos a simplificação pelos "Ø",

agrupamos quatro somas na primeira linha, simplificando as variáveis A e B, e quatro somas no grupo central, simplificando as "somas" C e $\bar{A}$, de uma única variável.

Exemplo 4.5.5.3:

Simplificar a função

$$Y = C.B.A + \bar{C}.B.\bar{A} + \bar{C}.B.A$$

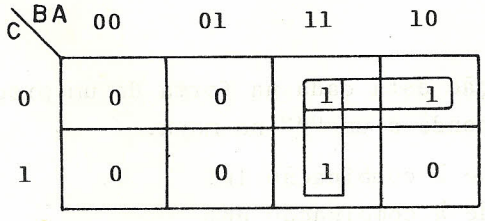
pelo mapa de Karnaugh e expressá-la

1º) como soma de produtos,

2º) como produto de somas.

Solução:

Os termos C.B.A, $\bar{C}$.B.$\bar{A}$ e $\bar{C}$.B.A correspondem, respectivamente, às combinações 111, Ø1Ø e Ø11. Plotando a função sobre o mapa e simplificando pelos "1"s, obtemos a função como uma soma de produtos, conforme ilustrado na fig. 4.53:

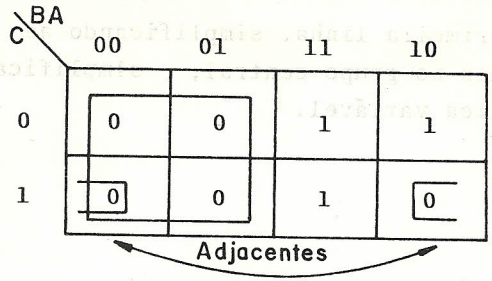


$Y = \bar{C}.B+B.A$

Fig. 4.53 - Mapa da Função do exemplo 4.5.5.3 e simplificação pelos "1"s.

Veja que este é o mesmo resultado do exemplo 4.5.4.1, onde a função foi simplificada algebricamente. Pelo mapa a simplificação é imediata.

A fig. 4.54 ilustra a simplificação pelos "Ø"s, com a função final na forma de um produto de somas.



$Y = (\overline{C}+A).B$

Fig. 4.54 - Mapa da Função do exemplo 4.5.5.3 e simplificação pelos "Ø"s.

Exemplo 4.5.54:

Simplificar a função

$Y = (\overline{C}+B+\overline{A}) \cdot (C+B+\overline{A}) \cdot (\overline{C}+\overline{B}+\overline{A})$

pelo mapa de Karnaugh e expressá-la:

1º) como um produto de somas,

2º) como uma soma de produtos.

Solução:

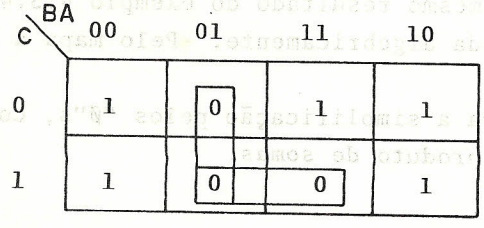
Neste caso, a função está dada na forma de um produto de somas e cada soma corresponde a um "Ø" no mapa.

$(\overline{C}+B+\overline{A})$ corresponde à combinação 1Ø1

$(C+B+\overline{A})$ corresponde à combinação ØØ1

$(\overline{C}+\overline{B}+\overline{A})$ corresponde à combinação 111

Plotando a função sobre o mapa e simplificando-a pelos "Ø"s obtemos a fig. 4.55:

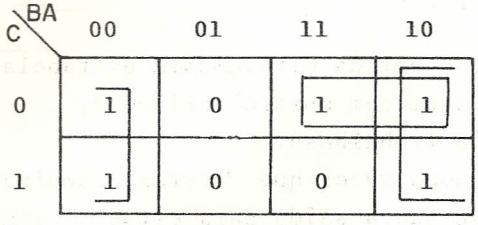


$Y = (B+\overline{A}).(\overline{C}+\overline{A})$

Fig. 4.55 - Mapa da Função do exemplo 4.5.54 e simplificação pelos "Ø"s.

Observe que este é o mesmo resultado obtido no exemplo 4.5.4.2, neste caso com muito menos trabalho.

Para conseguirmos a simplificação na forma de uma soma de produtos, basta agruparmos os termos iguais a "1", conforme ilustra a fig. 4.56:



$Y = \overline{A}+\overline{C}.B$

Fig. 4.56 - Mapa da Função do exemplo 4.5.5.4 e simplificação pelos "1"s.

Exemplo 4.5.5.5:

Simplificar a função plotada no mapa de Karnaugh da fig. 4.57, expressando-a na forma de uma soma de produtos.

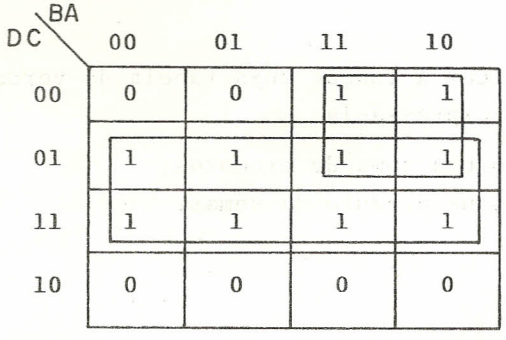


Fig. 4.57 - Mapa da Função do exemplo 4.5.5.5 e simplificação pelos "1"s.

Solução:

Conforme indicado na fig. 4.57, podemos agrupar os termos iguais a "1" em um óctuplo e em uma quadra, simplificando 3

e 2 variáveis, respectivamente. No óctuplo apenas a variável "C" permanece sempre em "1", correspondendo ao termo "C" na soma; na quadra das variáveis "D" e "B" permanecem iguais a "Ø" e "1", respectivamente, correspondendo à soma $\bar{D}.B$. Assim, a função simplificada fica:

$$Y = C + \bar{D} \cdot B$$

Observe que a função obtida diretamente da tabela da verdade teria 12 termos, cada um com quatro variáveis, e sua simplificação algébrica seria trabalhosa.

Pode acontecer algumas vezes que certas combinações das entradas nunca ocorram ou que a saída para estas combinações possa ser indiferentemente "Ø" ou "1". Nestes casos, podemos atribuir à saída qualquer valor, conforme seja mais conveniente para a simplificação. As combinações para as quais isto ocorre denominam-se condições irrelevantes (DON'T CARE) e a saída é indicada por um "X" na tabela da verdade ou mapa de Karnaugh.

Exemplo 4.5.5.6:

Simplificar a função cuja tabela da verdade é mostrada na fig. 4.58 e expressá-la:

1º) como uma soma de produtos,

2º) como um produto de somas.

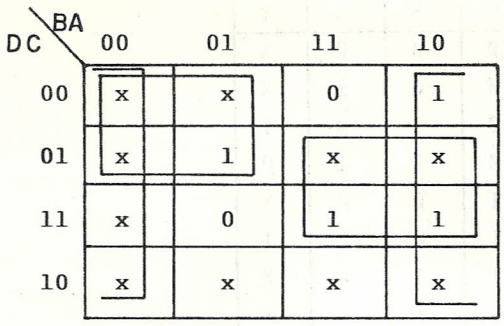
D	C	B	A	Y
Ø	Ø	Ø	Ø	X
Ø	Ø	Ø	1	X
Ø	Ø	1	Ø	1
Ø	Ø	1	1	Ø
Ø	1	Ø	Ø	X
Ø	1	Ø	1	1
Ø	1	1	Ø	X
Ø	1	1	1	X
1	Ø	Ø	Ø	X
1	Ø	Ø	1	X
1	Ø	1	Ø	X
1	Ø	1	1	X
1	1	Ø	Ø	X
1	1	Ø	1	Ø
1	1	1	Ø	1
1	1	1	1	1

Fig. 4.58 - Tabela da Verdade para a Função do exemplo 4.5.5.6

Solução:

Plotamos a função sobre o mapa de Karnaugh e agrupamos os termos iguais a "1" para obtermos a simplificação na forma de uma soma de produtos. Os termos irrelevantes podem ser considerados iguais a "1" ou "Ø", conforme for mais conveniente. Ao incluir um termo irrelevante num grupo de "1's", estabelecemos implicitamente que ele resultará igual a "1" na função final.

A fig. 4.59 ilustra a simplificação:



$Y = \overline{A}+C.B+\overline{D}.\overline{B}$

Fig. 4.59 - Mapa da Função do exemplo 4.5.5.6 e simplificação pelos "1"s.

A tabela da verdade para a função

$$Y = \bar{A} + C.B + \bar{D}.\bar{B}$$

está mostrada na fig. 4.60:

D	C	B	A	Y	
Ø	Ø	Ø	Ø	1	
Ø	Ø	Ø	1	1	
Ø	Ø	1	Ø	1	←
Ø	Ø	1	1	Ø	←
Ø	1	Ø	Ø	1	
Ø	1	Ø	1	1	←
Ø	1	1	Ø	1	
Ø	1	1	1	1	
1	Ø	Ø	Ø	1	
1	Ø	Ø	1	Ø	
1	Ø	1	Ø	1	
1	Ø	1	1	Ø	
1	1	Ø	Ø	1	
1	1	Ø	1	Ø	←
1	1	1	Ø	1	←
1	1	1	1	1	←

Fig. 4.60 - Tabela da Verdade para $Y=\overline{A}+C.B+\overline{D}.\overline{B}$.

Observe que, para as combinações assinaladas, a tabe

la da fig. 4.60 concorda exatamente com a da fig. 4.58, como não podia deixar de ser. As outras combinações são irrelevantes e, portanto, a saída resultante pode ser "Ø" ou "1".

A simplificação pelos "Ø"s está mostrada na fig.4.61:

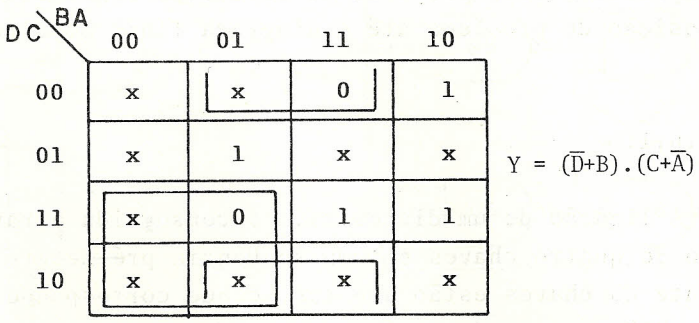


Fig. 4.61 - Mapa da Função do exemplo 4.5.5.6 e simplificação pelos "Ø"s.

A tabela da verdade para a função

$$Y = (\bar{D}+B) \cdot (C+\bar{A})$$

é mostrada na fig. 4.62. Observe que para os valores assinalados ela concorda com a tabela da fig. 4.58, sendo as demais combinações irrelevantes.

D	C	B	A	Y	
Ø	Ø	Ø	Ø	1	
Ø	Ø	Ø	1	Ø	
Ø	Ø	1	Ø	1	←
Ø	Ø	1	1	Ø	←
Ø	1	Ø	Ø	1	
Ø	1	Ø	1	1	←
Ø	1	1	Ø	1	
Ø	1	1	1	1	
1	Ø	Ø	Ø	Ø	
1	Ø	Ø	1	Ø	
1	Ø	1	Ø	1	
1	Ø	1	1	Ø	
1	1	Ø	Ø	Ø	
1	1	Ø	1	Ø	←
1	1	1	Ø	1	←
1	1	1	1	1	←

Fig. 4.62 - Tabela da Verdade para $Y=(\bar{D}+B) \cdot (C+\bar{A})$.

4.6 - Exemplos de Aplicação

Utilizando as técnicas expostas nos itens anteriores, vamos exemplificar dois projetos de circuitos combinacionais, desde a proposição do problema até o diagrama final do circuito.

Exemplo 4.6.1:

A ligação de um dispositivo é conseguida através do fechamento de quatro chaves em uma seqüência pré-determinada. Inicialmente as chaves estão abertas, o que corresponde a um nível lógico "1" na entrada do circuito de controle. Se uma das chaves for fechada fora da seqüência, será gerado um sinal ativo em nível "1" que provocará o bloqueio do dispositivo a ser acionado e acender-se-á um LED indicando o erro. Este LED exige 25 mA para acender-se. Se a seqüência correta for seguida, quando todas as chaves estiverem fechadas será gerado um sinal em nível lógico "1" que permitirá a ligação do dispositivo. Deseja-se implementar o circuito com portas lógicas CMOS, operando com alimentação Vcc = 10V.

Solução:

Vamos chamar de D, C, B e A as variáveis de entrada de nosso circuito, que representam o estado das chaves; a variável em "1" indica que a chave correspondente está aberta e em "Ø" que ela está fechada. Chamaremos "BL" a variável de saída que indica o bloqueio do dispositivo sob controle e, "LG", a variável de saída que indica a sua ligação.

Inicialmente todas as chaves estão abertas (D=C=B=A=1) e os sinais de bloqueio "BL" e ligação estão inativos (BL=LG=Ø). Vamos arbitrar que a seqüência desejada é: primeiro a chave D, segundo a chave C, terceiro a chave B e quarto a chave A. Se esta seqüência de fechamento for seguida, BL ficará inativo e LG tornar-se-á ativo quando as quatro variáveis forem iguais a

"Ø". Este resultado está expresso na fig. 4.63:

D	C	B	A	BL	LG
1	1	1	1	Ø	Ø
Ø	1	1	1	Ø	Ø
Ø	Ø	1	1	Ø	Ø
Ø	Ø	Ø	1	Ø	Ø
Ø	Ø	Ø	Ø	Ø	1

Fig. 4.63 - Seqüência de Fechamento das Chaves e Sinais de Controle para o exemplo 4.6.1.

Todas as outras combinações das variáveis de entrada provocarão o ativamento do sinal de bloqueio BL, que irá para "1". Quando BL for para "1", deve acender-se o LED indicando o erro. O sinal que permite a ligação do dispositivo sob controle só ficará ativo ao final da seqüência, quando as quatro variáveis de entrada estiverem em "Ø". A fig. 4.64 mostra a tabela da verdade para o circuito de controle:

D	C	B	A	BL	LG
Ø	Ø	Ø	Ø	Ø	1
Ø	Ø	Ø	1	Ø	Ø
Ø	Ø	1	Ø	1	Ø
Ø	Ø	1	1	Ø	Ø
Ø	1	Ø	Ø	1	Ø
Ø	1	Ø	1	1	Ø
Ø	1	1	Ø	1	Ø
Ø	1	1	1	Ø	Ø

D	C	B	A	BL	LG
1	Ø	Ø	Ø	1	Ø
1	Ø	Ø	1	1	Ø
1	Ø	1	Ø	1	Ø
1	Ø	1	1	1	Ø
1	1	Ø	Ø	1	Ø
1	1	Ø	1	1	Ø
1	1	1	Ø	1	Ø
1	1	1	1	Ø	Ø

Fig. 4.64 - Tabela da Verdade para o Circuito de Controle.

De imediato vê-se que a equação para a saída LG é:

$$LG = (\overline{A + B + C + D}) = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}$$

Para "BL" plotamos a função no mapa de Karnaugh e simplificamos pelos "1"s ou "Ø"s. A simplificação pelos "1"s está ilustrada na fig. 4.65:

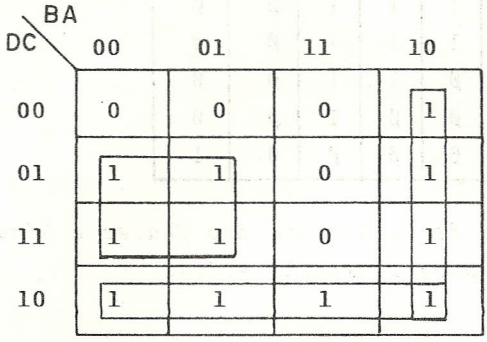


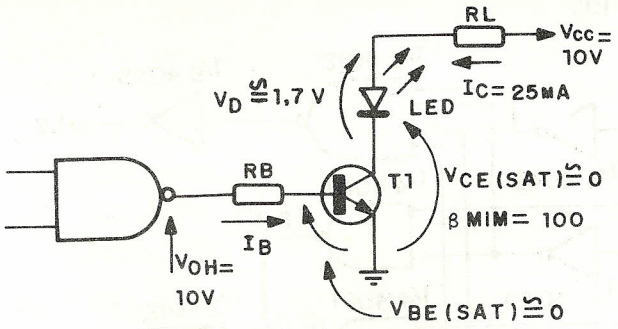
Fig. 4.65 - Mapa de Karnaugh e Simplificação da Variável de saída "BL".

Assim, as equações para as variáveis de saída do circuito de controle são:

$$BL = C.\bar{B} + D.\bar{C} + B.\bar{A}$$

$$LG = \bar{D}.\bar{C}.\bar{B}.\bar{A}$$

Desejamos ainda que, com "BL" em "1", se acenda o LED. Um diodo emissor de luz apresenta, em geral, uma tensão direta em torno de 1,7V e, como a saída da porta no nível lógico "1" será próxima de, Vcc = 10V, poderíamos pensar em lugar o LED diretamente à saída da porta através de um resistor. Entretanto, não podemos fazer isto, porque a saída de uma porta lógica CMOS no estado "1" é capaz de fornecer apenas 1 mA (parâmetro I_{OH}, veja o capítulo 3). Devemos colocar um circuito de interface entre a saída da porta e o LED para fornecer-lhe os 25 mA de que necessita. Este circuito poderia ser um transistor, conforme ilustra a fig. 4.66, onde se indica o cálculo das correntes, tensões e resistências envolvidas quando a saída da porta lógica estiver no nível "1".



$$R_L = \frac{V_{CC}-V_D}{I_C} = \frac{10-1,7}{25} \simeq 0,33K\Omega$$

$$R_B = \frac{V_{OH}}{(I_C/\beta \text{ min})} = 40K\Omega$$

Fig. 4.66 - Interface da Porta Lógica CMOS com o LED de 25 mA.

No circuito da fig. 4.66, utilizamos um transistor T_1 com β min = 1ØØ, de maneira que a porta lógica deve fornecer apenas Ø,25 mA (Ic/βmin) à sua base para que ele entre na saturação. Com o transistor saturado, a tensão V_{CE} será muito pequena e a corrente de coletor estará limitada pela resistência, cujo valor é dimensionado de modo a permitir a circulação de 25 mA pelo LED, cuja tensão direta foi estimada em 1,7V.

A implementação do circuito de controle a partir das equações que obtivemos anteriormente pode ser feita de diversas maneiras. A fig. 4.67 ilustra uma das formas possíveis, utilizando exclusivamente portas NAND e inversores.

Neste circuito os sinais de entrada D, C, B e A provêm das chaves S_3, S_2, S_1 e $S_Ø$, respectivamente.

Observe que, se mudarmos os JUMPERS mostrados na figura, poderemos selecionar qualquer das seqüências de fechamento possíveis para as chaves.

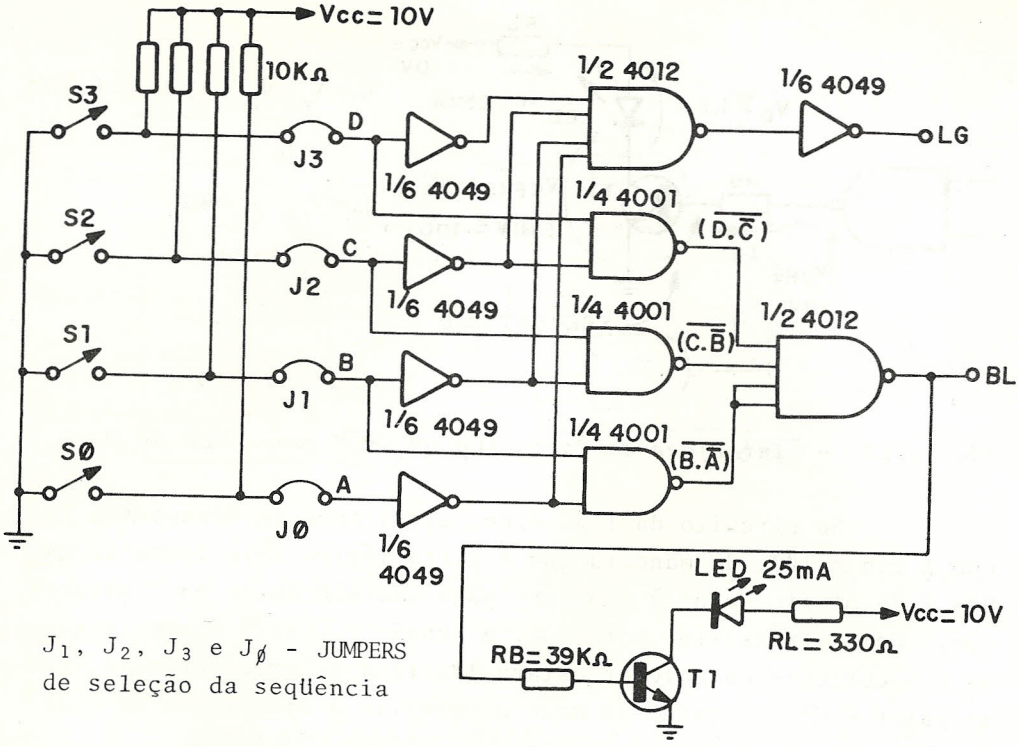


Fig. 4.67 - Circuito Implementado com portas NAND e inversores CMOS.

Veja no circuito da fig. 4.67 a utilização de uma das portas NAND 4012 como de três entradas. É conveniente procurar utilizar ao máximo os CI'S disponíveis, para reduzir o custo evitando a introdução de circuitos integrados desnecessários, como seria o caso se não usássemos esta porta e, em seu lugar, colocássemos 1/3 4023, por exemplo.

Exemplo 4.6.2:

Um circuito recebe dois números binários "A" e "B" de quatro bits e deve fornecer três sinais de saída indicando a grandeza relativa dos números. Este sinais de saída são A>B,

A=B e A<B e somente um deles deve ficar ativo no nível lógico "1", de acordo com o resultado da comparação. Deseja-se implementar o circuito com portas lógicas da família TTL.

Solução:

Vamos chamar os quatro bits do número "A" de A_3, A_2, A_1 $A_\emptyset$, sendo A_3 o bit mais significativo ou MSB(MOST SIGNIFICANT BIT), e os quatro bits do número "B" de B_3, B_2, B_1 e $B_\emptyset$ (B_3=MSB).

Para resolver este problema, poderíamos pensar em considerar B_3, B_2, B_1, $B_\emptyset$, A_3, A_2, A_1 e $A_\emptyset$ como variáveis de entrada para as funções A>B, A=B e A<B e tentar fazer a tabela da verdade e os mapas de Karnaugh para as saídas. No entanto, imagine o que é simplificar uma função de oito variáveis de entrada! Neste caso, temos que procurar outra solução.

Consideremos dois bits quaisquer An e Bn. Para apenas estes dois bits é fácil fazer a comparação, conforme mostra a tabela da verdade apresentada na fig. 4.68:

An	Bn	An > Bn	An = Bn	An < Bn
Ø	Ø	Ø	1	Ø
Ø	1	Ø	Ø	1
1	Ø	1	Ø	Ø
1	1	Ø	1	Ø

Fig. 4.68 - Tabela da Verdade para Comparação de dois bits An e Bn.

Da tabela da verdade tiramos imediatamente que:

$$(An > Bn) = An.\overline{Bn}$$

$$(An = Bn) = \overline{An} \cdot \overline{Bn} + An \cdot Bn = \overline{An \oplus Bn}$$

$$(An < Bn) = \overline{An} \cdot Bn = \overline{(An > Bn)} \cdot \overline{(An = Bn)}$$

Observe que a função (An=Bn) é a nossa conhecida fun

ção Equivalência ou EXCLUSIVE-NOR ou ainda o OU-EXCLUSIVO invertido.

Pensemos agora, em termos dos dois números "A" e "B" O número "A" será maior que o número "B" se

1) $A_3 > B_3$, "OU"
2) $A_3 = B_3$ "E" $A_2 > B_2$, "OU"
3) $A_3 = B_3$ "E" $A_2 = B_2$ "E" $A_1 > B_1$, "OU"
4) $A_3 = B_3$ "E" $A_2 = B_2$ "E" $A_1 = B_1$ "E" $A_\emptyset > B_\emptyset$

O número "A" será igual ao número "B" se, e somente se

$$A_3 = B_3 \text{ "E" } A_2 = B_2 \text{ "E" } A_1 = B_1 \text{ "E" } A_\emptyset = B_\emptyset$$

e, finalmente, o número "A" será menor que o número "B" se não for nem maior nem igual.

Podemos resumir este raciocínio nas equações:

$$(A>B) = (A_3>B_3) + (A_3=B_3) \cdot (A_2>B_2) + (A_3=B_3) \cdot (A_2=B_2) \cdot (A_1>B_1) + (A_3=B_3) \cdot (A_2=B_2) \cdot (A_1=B_1) \cdot (A_\emptyset>B_\emptyset)$$

$$(A=B) = (A_3=B_3) \cdot (A_2=B_2) \cdot (A_1=B_1) \cdot (A_\emptyset=B_\emptyset)$$

$$(A<B) = (\overline{A>B}) \cdot (\overline{A=B})$$

O comparador de dois bits An e Bn pode ser implementado pelo circuito mostrado na fig. 4.69, fornecendo as saídas An > Bn e An = Bn :

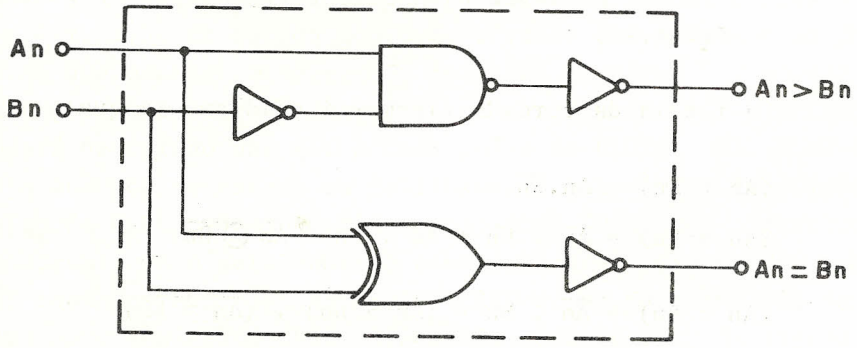


Fig. 4.69 - Comparador para dois Bits An e Bn.

O comparador para os dois números de quatro bits utilizaria blocos como o da fig. 4.69, interligados conforme mostra a fig. 4.70:

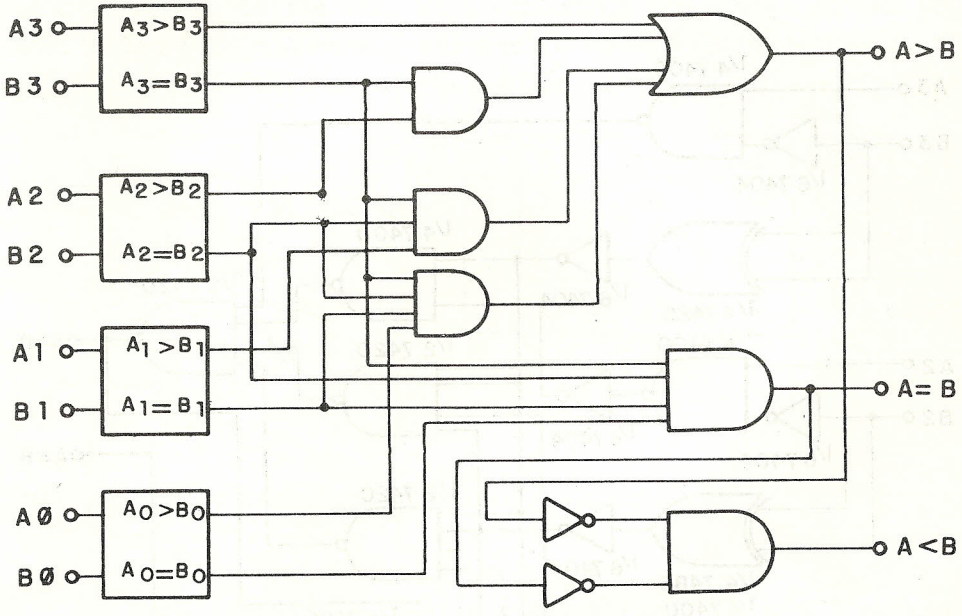
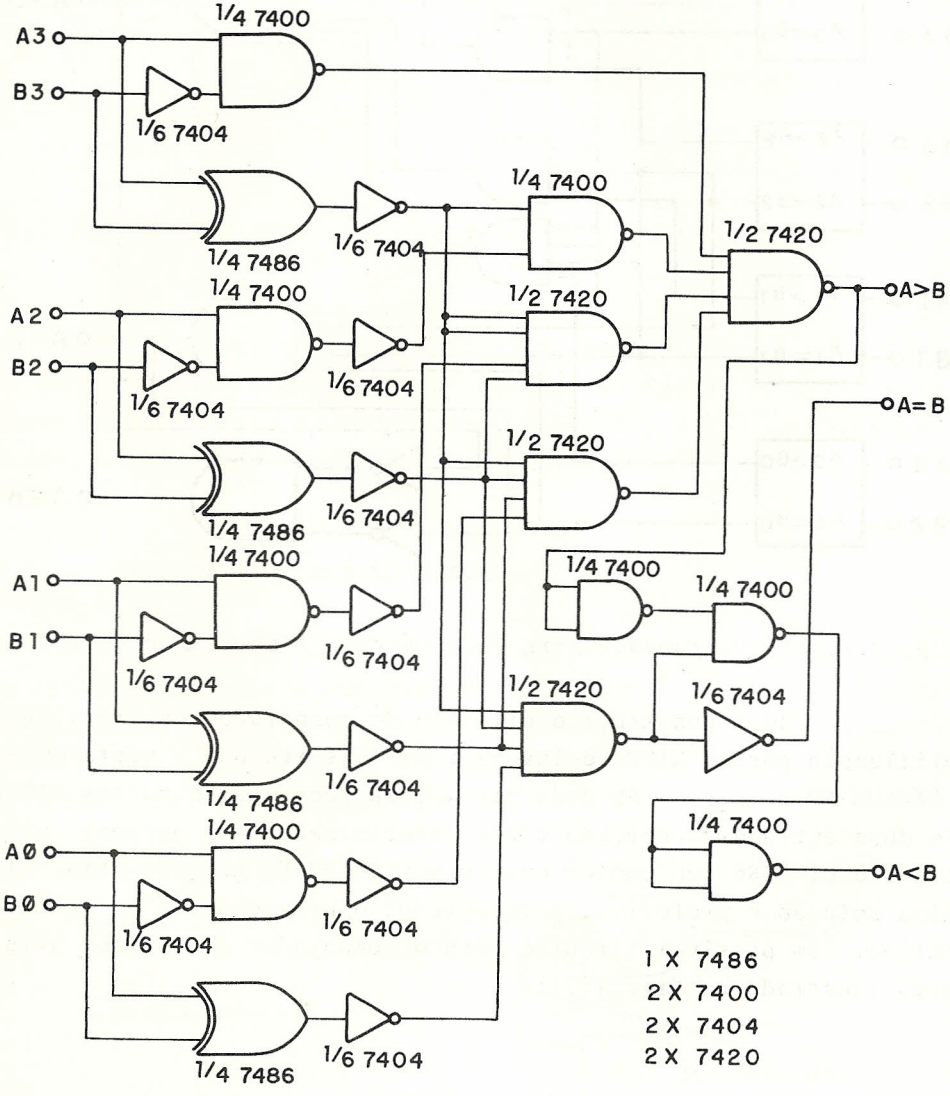


Fig. 4.70 - Comparador para Dois números Binários de 4 bits.

Vamos construir o circuito do comparador da fig.4.70, utilizando portas NAND e inversores. Observe que a porta OU-EXCLUSIVO da fig. 4.69 pode ser obtida com quatro portas NAND de duas entradas, conforme vimos anteriormente, ou podemos utilizar o CI 7486 que contém quatro portas OU-EXCLUSIVO. Esta última solução é preferível porque reduz o número de CI's necessários. Um possível circuito para o comparador de quatro bits está ilustrado na fig. 4.71:


A3
B3
A2
B2
A1
B1
A0
B0
1/4 7400
1/6 7404
1/4 7486
1/2 7420
A>B
A=B
A<B
1 X 7486
2 X 7400
2 X 7404
2 X 7420

CAPÍTULO 5 - FLIP-FLOPS

5.1 - Introdução

O FLIP-FLOP ou multivibrador biestável é um circuito que possui apenas dois estados estáveis, podendo permanecer indefinidamente em qualquer deles. A figura 5.1 mostra o esquema de um FLIP-FLOP construído com elementos discretos, apenas para ilustrar o conceito dos estados estáveis:

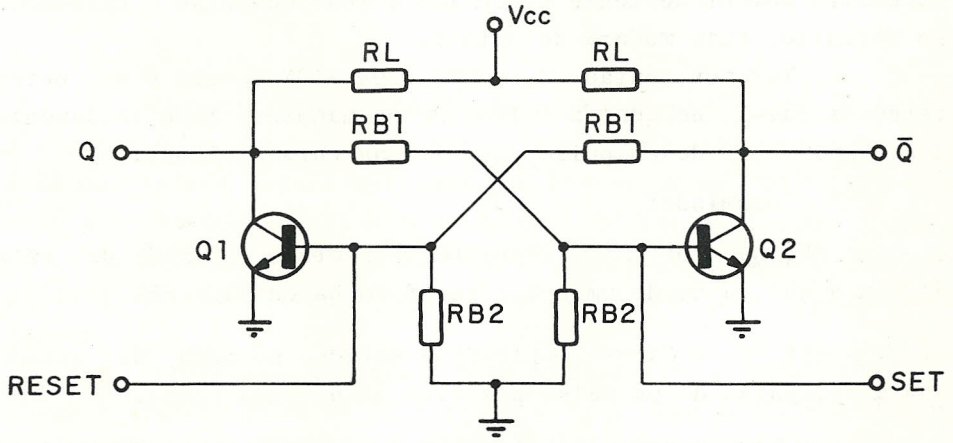


Fig. 5.1 - FLIP-FLOP RS com Componentes Discretos.

Este circuito é projetado de maneira que ocorra o seguinte:

- ao se aplicar a tensão de alimentação, um dos transistores conduzirá mais do que o outro; vamos supor que este seja Q_1;
- a maior condução de Q_1 fará com que sua tensão de coletor diminua, reduzindo a corrente de base de Q_2;
- com menor corrente de base, Q_2 conduzirá menos e sua tensão de coletor aumentará, provocando um aumento na corrente de base de Q_1;
- Q_1 conduzirá mais, indo para a saturação, enquanto Q_2 irá pa

ra o corte.

O processo acima descrito é muito rápido e, em alguns microssegundos, será atingido um estado estável, correspondente a Q_1 saturado (ON) e Q_2 cortado (OFF).

Uma vez atingido este estado, o FLIP-FLOP se manterá indefinidamente nele. As saídas de um FLIP-FLOP são designadas convencionalmente Q e $\overline{Q}$, sendo complementares. Quando Q = "Ø" e $\overline{Q}$ = "1" dizemos que o FLIP-FLOP está ressetado, caso contrário, dizemos que o mesmo está setado.

Estando Q_1 = ON e Q_2 = OFF no circuito da figura 5.1, se aplicarmos um pulso positivo à entrada SET do FLIP-FLOP, Q_2 conduzirá momentaneamente e, devido à realimentação existente no circuito, este mudará de estado.

Teremos, então, Q_1 = OFF e Q_2 = ON e este é o outro estado estável, no qual o FLIP-FLOP permanecerá indefinidamente até a aplicação de um pulso positivo na entrada RESET.

Resumindo:

1) Q_1 = ON, Q_2 = OFF ⟹ FLIP-FLOP ressetado, só muda de estado com a aplicação de um pulso positivo na entrada SET.

2) Q_1 = OFF, Q_2 = ON ⟹ FLIP-FLOP setado, só muda de estado com a aplicação de um pulso positivo na entrada RESET.

Observe que este circuito é capaz de armazenar uma informação, qual seja a ocorrência de um pulso na linha SET ou na linha RESET.

Os FLIP-FLOPS são o elemento básico na construção de circuitos seqüenciais, como contadores, registradores de deslocamento, etc. Neste capítulo, examinaremos os diversos tipos de FLIP-FLOPS existentes na forma de CI'S,apresentando suas características e aplicações.

5.2 - FLIP-FLOP com Portas Lógicas

Um Circuito análogo ao da figura 5.1 pode ser construído utilizando portas lógicas. A figura 5.2 ilustra o circuito básico de um FLIP-FLOP construído com portas NAND, também co

nhecido com o nome de $\overline{R}$-$\overline{S}$ NAND LATCH:

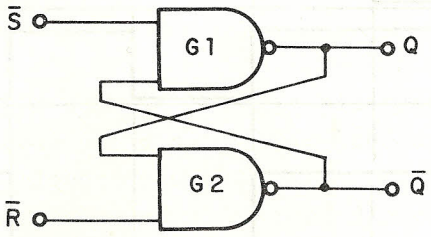


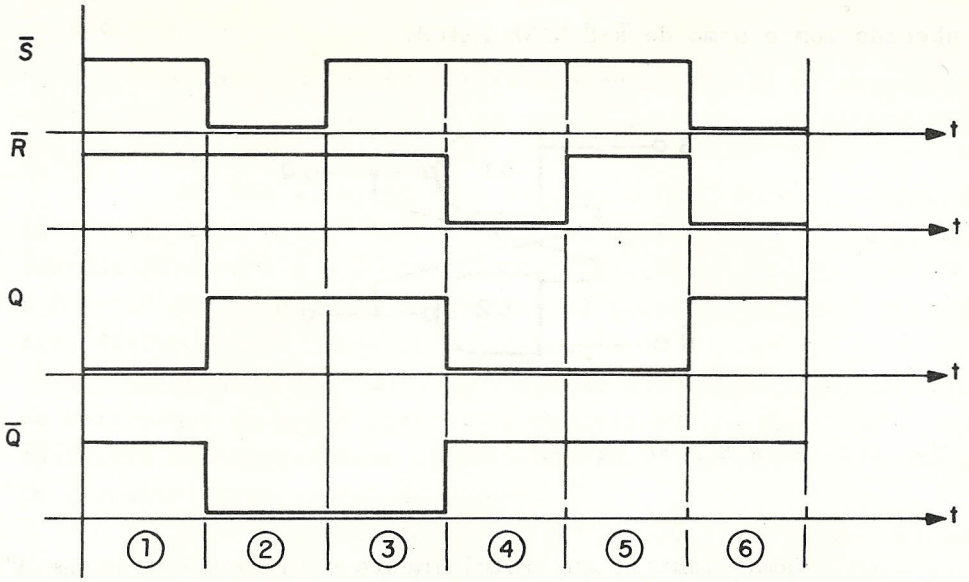
Fig. 5.2 - $\overline{R}$ $\overline{S}$ NAND LATCH.

Vamos admitir que inicialmente a saída Q esteja em "Ø" e a saída $\overline{Q}$ em "1", ou seja que o FLIP-FLOP esteja ressetado. Suponhamos também que as entradas $\overline{S}$ e $\overline{R}$ estejam em "1". O FLIP-FLOP está num estado estável e a saída Q = "Ø" força a saída $\overline{Q}$ para "1". Se nesta situação a entrada $\overline{S}$ for para "Ø" com $\overline{R}$ = "1", a saída Q irá para "1" e a saída $\overline{Q}$ para "Ø" e o FLIP-FLOP estará setado.

Mesmo que a entrada $\overline{S}$ volte para "1", o FLIP-FLOP continuará setado, pois $\overline{Q}$="Ø" força Q = "1". Com o FLIP-FLOP setado (Q ="1", $\overline{Q}$ = "Ø"), se a entrada $\overline{R}$ for para "Ø", sendo $\overline{S}$="1", $\overline{Q}$ irá para "1" e Q para "Ø"; o FLIP-FLOP será ressetado e se manterá assim, mesmo que $\overline{R}$ volte para "1".

Caso $\overline{R}$ e $\overline{S}$ fiquem simultaneamente em "Ø", Q e $\overline{Q}$ ficarão em "1"; isto contraria nossa definição de Q e $\overline{Q}$ como saídas complementares e assim a combinação $\overline{R}$=$\overline{S}$="Ø" não é permitida.

A figura 5.3 mostra as formas de onda das entradas e saídas do circuito da fig. 5.2, ilustrando e esclarecendo a discussão acima.



1) FLIP-FLOP ressetado, $\overline{R}$ e $\overline{S}$ em "1"
2) $\overline{S}$ vai para "Ø", o FLIP-FLOP é setado
3) $\overline{S}$ volta para "1", o FLIP-FLOP permanece setado
4) $\overline{R}$ vai para "Ø", o FLIP-FLOP é ressetado
5) $\overline{R}$ volta para "1", o FLIP-FLOP permanece ressetado
6) $\overline{S}$ e $\overline{R}$ para "Ø", Q e $\overline{Q}$ ficam em "1", contrariando a definição.

Fig. 5.3 - Formas de Onda Ilustrativas do Funcionamento do $\overline{R}$ - $\overline{S}$ NAND LATCH.

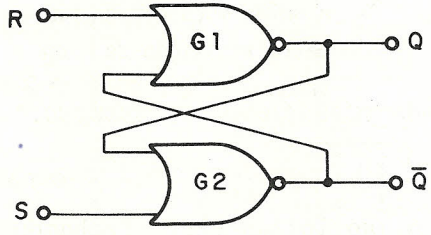
As entradas $\overline{S}$ e $\overline{R}$ são denominadas SET e RESET, respectivamente . A tabela da verdade do FLIP-FLOP $\overline{R}$ $\overline{S}$ está mostrada na figura 5.4:

R	S	Q n+1	$\overline{Q}$ n+1	OBS:
Ø	Ø	1	1	não permitido
Ø	1	Ø	1	ressetado
1	Ø	1	Ø	setado
1	1	Qn	$\overline{Q}$n	não muda

Fig. 5.4 - Tabela da Verdade para o $\overline{R}$ $\overline{S}$ NAND LATCH.

Nesta tabela Qn e $\overline{Qn}$ representam o estado presente das saídas e Qn+1, $\overline{Qn+1}$ o estado que assumem como resposta às variações das entradas.

A figura 5.5 mostra o circuito básico de um FLIP-FLOP construído com portas NOR, mais conhecido com o nome de RS NOR LATCH. Seu funcionamento é análogo ao do $\overline{R}\ \overline{S}$ NAND LATCH. A tabela da verdade correspondente também está mostrada na fig.5.5:



R	S	Qn+1	$\overline{Qn+1}$
Ø	Ø	Qn	$\overline{Qn}$
Ø	1	1	Ø
1	Ø	Ø	1
1	1	Ø	Ø

OBS:
não muda
setado
ressetado
não permitido.

Fig. 5.5 - RS NOR LATCH e sua Tabela da Verdade.

Os LATCHES apresentados nas figuras 5.2 e 5.5 possuem uma aplicação importante em circuitos de DEBOUNCING (ou eliminação de rebote). Quando usamos uma chave para gerar um sinal de controle para um circuito lógico, pode ocorrer a geração de sinais espúrios no momento da abertura ou fechamento de seus contatos. Usamos, neste caso, um LATCH para "limpar" o sinal, aproveitando o fato de que o circuito mudará de estado com a primeira transição de sua entrada e não responderá às variações subseqüentes. A figura 5.6 mostra esta aplicação para um LATCH:

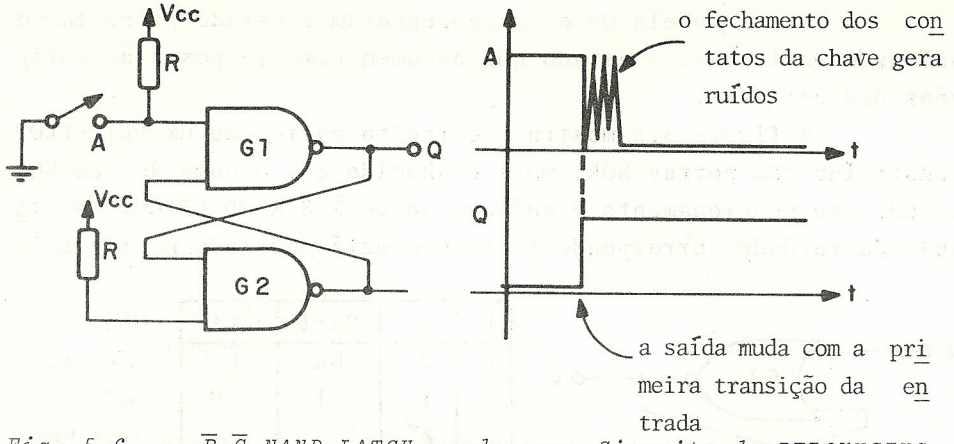


Fig. 5.6 - $\overline{R}\ \overline{S}$ NAND LATCH usado como Circuito de DEBOUNCING.

Conforme ilustra a fig. 5.6, no instante do fechamento da chave ocorre a geração de ruídos, devido a problemas mecânicos no cerramento dos contatos (BOUNCING). O LATCH elimina os ruídos e entrega ao restante do circuito um sinal limpo.

5.3 - FLIP-FLOP RS MASTER-SLAVE

Na maioria dos circuitos digitais deseja-se que todas as mudanças de estado ocorram em sincronismo. O sinal utilizado para sincronizar o funcionamento do circuito é denominado CLOCK. A fig. 5.7 ilustra um FLIP-FLOP RS que utiliza o CLOCK para habilitar a mudança de estado:

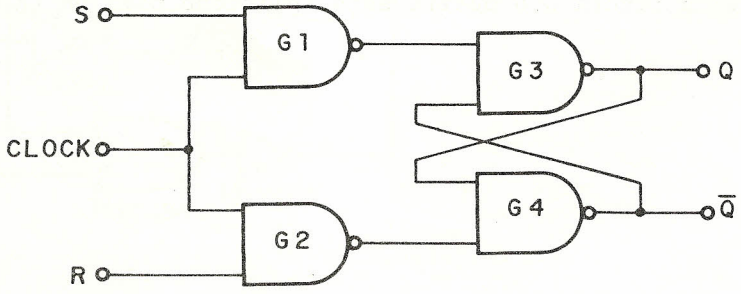
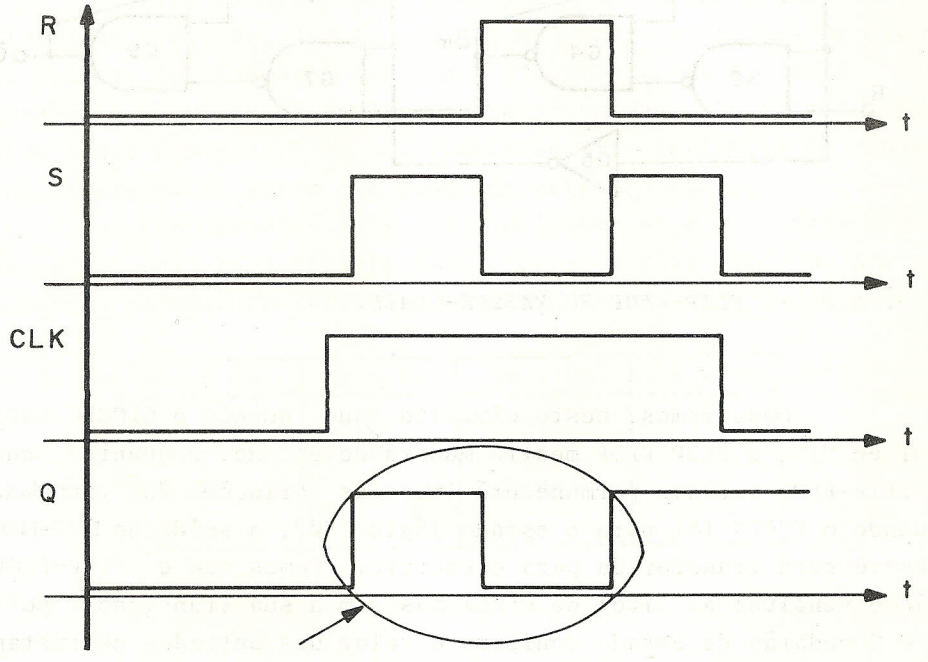


Fig. 5.7 - RS NAND LATCH com CLOCK.

No circuito da fig. 5.7, se o CLOCK estiver em "Ø" as saídas Q e $\overline{Q}$ permanecerão insensíveis às variáveis das entradas. Somente quando o CLOCK estiver no estado lógico "1" é que as variações das entradas se refletirão nas saídas.

Este circuito é sensível ao nível do CLOCK, ou seja, estando o CLOCK em "1", as saídas acompanharão as entradas. Isto significa que, se as entradas variam durante o tempo em que o CLOCK as habilita, a saída mudará de estado mais de uma vez durante o pulso de CLOCK, o que não é, em geral, desejável. A figura 5.8 ilustra este ponto com algumas formas de onda:



a saída mudou de estado duas vezes durante um único pulso de CLOCK.

Fig. 5.8 - Comportamento do Circuito da fig. 5.7 quando as Entradas variam durante o Pulso de Clock.

Para contornarmos este problema, utilizamos dois estágios como o da fig. 5.7 em cascata, com seus CLOCKS invertidos.

O primeiro estágio se denomina mestre e, o segundo, escravo, sendo a configuração conhecida por FLIP-FLOP mestre-escravo ou MASTER-SLAVE. A figura 5.9 mostra o FLIP-FLOP RS MASTER-SLAVE:

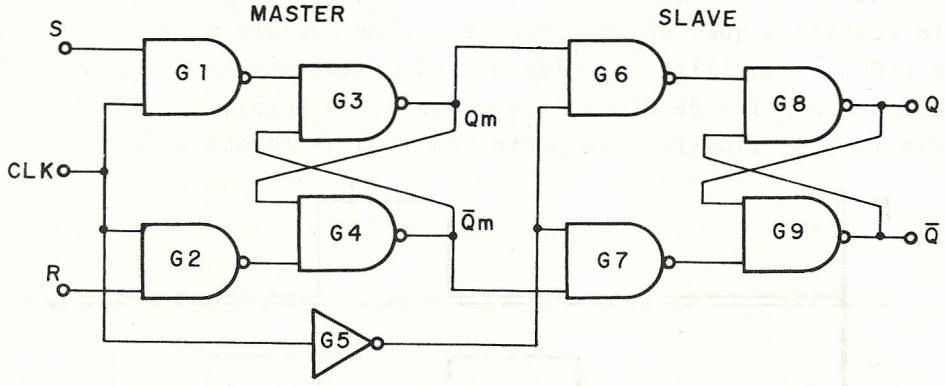


Fig. 5.9 - FLIP-FLOP RS MASTER-SLAVE.

Observamos, neste circuito, que, quando o CLOCK estiver em "1", o FLIP-FLOP mestre mudará de estado, enquanto que o FLIP-FLOP escravo permanecerá imune às variações das entradas. Quando o CLOCK for para o estado lógico "Ø", a saída do FLIP-FLOP mestre será transferida para o escravo. Vemos que o FLIP-FLOP não é sensível ao nível do CLOCK mas sim à sua transição, pois Q e $\bar{Q}$ mudarão de estado conforme o valor das entradas no instante em que ocorrer a *transição* do CLOCK do nível "1" para o nível "Ø". Este arranjo garante que a saída do FLIP-FLOP somente mudará de estado uma vez a cada pulso de CLOCK. Os FLIP-FLOPS cuja mudança de estado ou gatilhamento ocorram na transição do pulso de CLOCK são conhecidos como EDGE TRIGGERED (gatilhado pela borda); se o gatilhamento ocorre na transição de "Ø" para "1", são chamados POSITIVE EDGE-TRIGGERED e, se na transição de "1" para "Ø", NEGATIVE EDGE-TRIGGERED. O FLIP-FLOP da figura 5.9 é, então, do tipo NEGATIVE EDGE-TRIGGERED, ou gatilhado pela transição negativa de CLOCK.

Se as entradas variarem exatamente na transição do CLOCK, o gatilhamento do FLIP-FLOP será imprevisível. Torna-se necessário definir dois parâmetros que devem ser obedecidos para que o circuito funcione corretamente:

t_H - HOLD TIME, é o tempo que uma entrada deve permanecer estável após a transição do CLOCK que causa o gatilhamento do FLIP-FLOP;

t_S - SETUP TIME, é o tempo que uma entrada deve permanecer estável antes da transição do CLOCK que causa o gatilhamento do FLIP-FLOP.

A figura 5.10 ilustra a definição destes dois parâmetros:

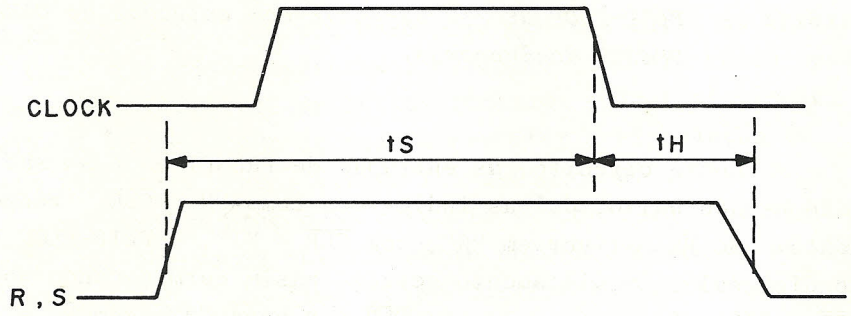


Fig. 5.10 - Definição de t_S e t_H para um FLIP-FLOP gatilhado na descida do CLOCK.

O circuito da figura 5.9 é um FLIP-FLOP RS MASTER-SLAVE gatilhado na transição negativa do CLOCK. Uma sofisticação adicional pode ser conseguida incluindo duas entradas de PRESET e CLEAR assíncronos, ou seja, independentes do CLOCK. O circuito resultante é mostrado na figura 5.11:

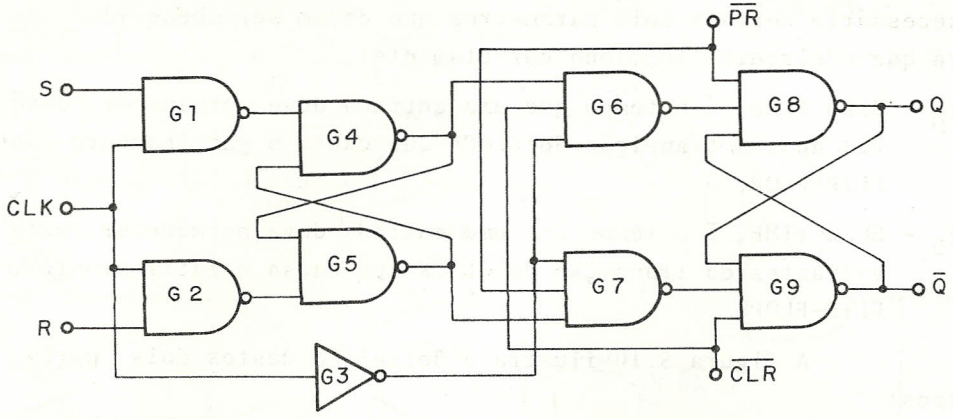


Fig. 5.11 - FLIP-FLOP RS MASTER-SLAVE com entradas de CLEAR e PRESET assíncronas.

Neste circuito, as entradas de PRESET e CLEAR são ativas no estado baixo, daí as indicarmos como $\overline{PR}$ e $\overline{CLR}$, respectivamente. Se $\overline{PR}$ estiver em "Ø", com $\overline{CLR}$ = "1", o FLIP-FLOP será incondicional e imediatamente setado, assim permanecendo enquanto $\overline{PR}$ = "Ø". Do mesmo modo, se $\overline{CLR}$ for para "Ø", com $\overline{PR}$ ="1" o FLIP-FLOP será imediatamente ressetado, assim permanecendo enquanto $\overline{CLR}$ = "Ø". As entradas diretas $\overline{CLR}$ e $\overline{PR}$ são, em geral, utilizadas somente no início ou fim de alguma seqüência, para colocar o FLIP-FLOP em um estado predeterminado; normalmente permanecem inativas durante a maior parte do tempo. Observe que somente uma das entradas diretas pode estar ativa em um dado momento, caso contrário, o FLIP-FLOP ficará numa condição não permitida, com $Q = \overline{Q} = 1$.

A figura 5.12 mostra a tabela da verdade para o circuito da figura 5.11:

$\overline{CLR}$	$\overline{PR}$	R	S	CLK	Qn+1	$\overline{Q}$n+1	OBS:
Ø	Ø	X	X	X	1	1	não permitida
Ø	1	X	X	X	Ø	1	ressetado (ass.)
1	Ø	X	X	X	1	Ø	setado (ass.)
1	1	Ø	Ø	↓	Qn	$\overline{Q}$n	não muda
1	1	Ø	1	↓	1	Ø	setado na descida de CLK
1	1	1	Ø	↓	Ø	1	ressetado na descida de CLK
1	1	1	1	↓	1	1	não permitida

Fig. 5.12 - Tabela da Verdade para o FLIP-FLOP da figura 5.11.

Na tabela da verdade acima o símbolo "↓" representa a transição negativa do pulso de CLOCK e "X" uma condição irrelevante.

5.4 - FLIP-FLOP JK MASTER-SLAVE

Vimos, no item anterior, que o FLIP-FLOP RS não permite a combinação R = S = "1" para as entradas, pois, neste caso, as saídas Q e $\overline{Q}$ deixariam de ser complementares, contrariando a definição. Esta dificuldade é superada com o FLIP-FLOP "JK", cujo circuito básico está ilustrado na fig. 5.13:

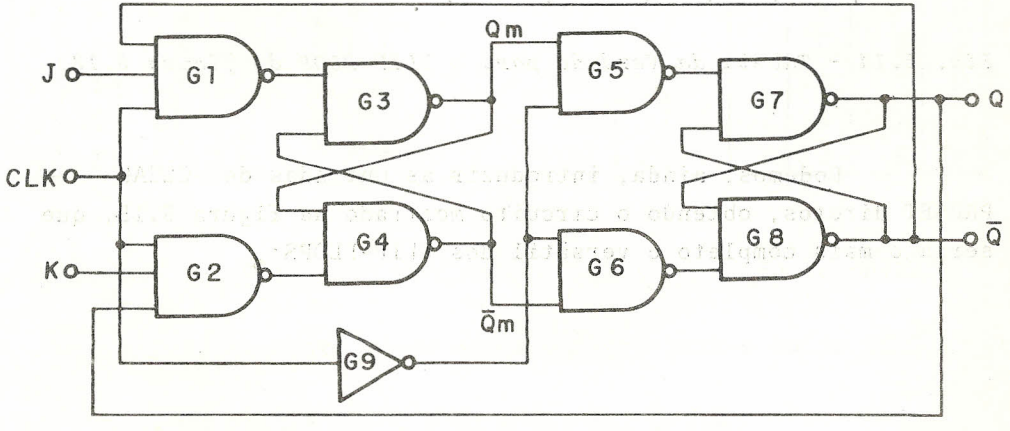


Fig. 5.13 - FLIP-FLOP JK MASTER-SLAVE gatilhado na transição negativa do CLOCK.

Vamos analisar as quatro combinações de "J" e "K" e seus efeitos.

1) J = "Ø", K = "Ø"

O FLIP-FLOP não muda de estado. Se estiver setado permanece setado e se estiver ressetado permanece ressetado.

2) J = "1", K = "Ø"

O FLIP-FLOP é setado. Se já estiver setado, não muda de estado.

3) J = "Ø", K = "1"

O FLIP-FLOP é ressetado. Se já estiver ressetado, não muda de estado.

4) J = "1", K = "1"

O FLIP-FLOP complementa seu estado (TOGGLE). Se estiver setado é ressetado e se estiver ressetado é setado.

A tabela da verdade mostrada na figura 5.14 resume o funcionamento do FLIP-FLOP J-K MASTER SLAVE da figura 5.13:

CLK	J	K	Qn+1	$\overline{Q}$n+1	OBS:
↓	Ø	Ø	Qn	$\overline{Q}$n	não muda
↓	Ø	1	Ø	1	é ressetado
↓	1	Ø	1	Ø	é setado
↓	1	1	$\overline{Q}$n	Qn	complementa (TOGGLE)

Fig. 5.14 - Tabela da Verdade para o FLIP-FLOP da figura 5.13.

Podemos, ainda, introduzir as entradas de CLEAR e PRESET diretos, obtendo o circuito mostrado na figura 5.15, que seria o mais completo e versátil dos FLIP-FLOPS:

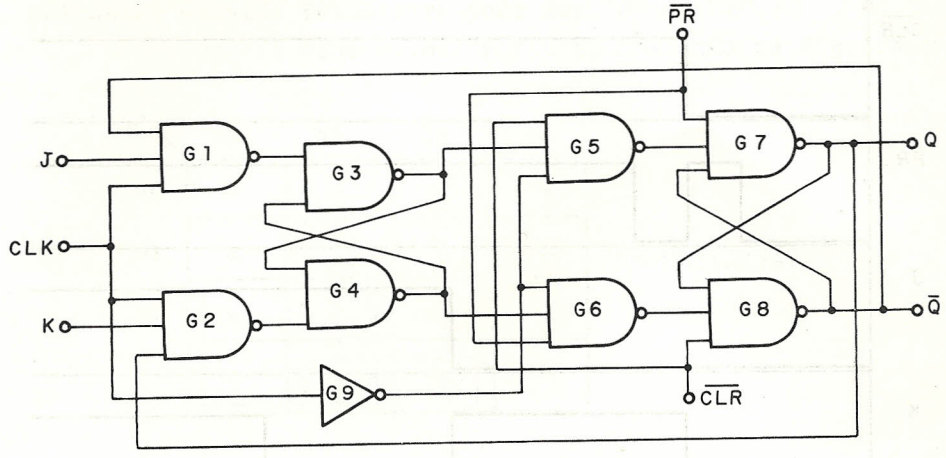


Fig. 5.15 - FLIP-FLOP JK MASTER-SLAVE com entradas de CLEAR e PRESET assíncronos.

A tabela da verdade para este circuito está ilustrada na figura 5.16:

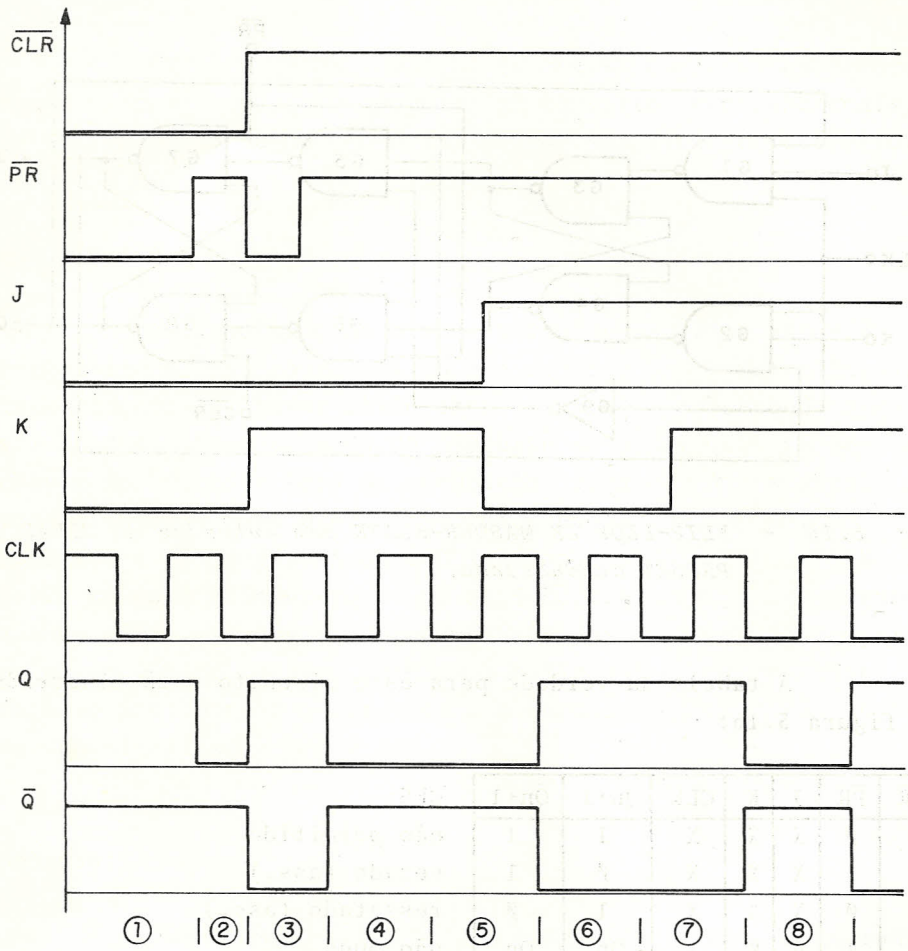
$\overline{CLR}$	$\overline{PR}$	J	K	CLK	Qn+1	$\overline{Q}$n+1
Ø	Ø	X	X	X	1	1
Ø	1	X	X	X	Ø	1
1	Ø	X	X	X	1	Ø
1	1	Ø	Ø	↓	Qn	$\overline{Q}$n
1	1	Ø	1	↓	Ø	1
1	1	1	Ø	↓	1	Ø
1	1	1	1	↓	$\overline{Q}$n	Qn

OBS:
não permitido
setado (ass.)
ressetado (ass.)
não muda
ressetado na trans. ↓ do CLK
setado na trans. ↓ do CLK
complementa na trans. ↓ do CLK.

Fig. 5.16 - Tabela da Verdade para o FLIP-FLOP JK MASTER-SLAVE da figura 5.15.

A figura 5.17 mostra algumas formas de onda ilustrativas do funcionamento do FLIP-FLOP da figura 5.15 e que esclare

cem a tabela da verdade da figura 5.15:



1) $\overline{CLR}$ = $\overline{PR}$ = "Ø", Q = $\bar{Q}$ = "1", condição não permitida

2) $\overline{PR}$ vai para "1", $\overline{CLR}$ = "Ø", o FF é imediatamente ressetado

3) $\overline{CLR}$ vai para "1", $\overline{PR}$ vai para "Ø", o FF é imediatamente setado.

4) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "Ø", K = "1", o FF é ressetado na transição ↓ do CLK.

5) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "Ø", K = "1", o FF permanece ressetado.

6) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "1", K = "Ø", o FF é setado na trans ↓ do CLK.

7) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "1", K = "Ø", o FF permanece setado.

8) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "1", K = "1", o FF complementa (neste caso, é ressetado) na transição negativa do CLOCK.

9) $\overline{CLR}$ = $\overline{PR}$ = "1", J = "1", K = "1", o FF complementa novamente (neste caso, é setado) na transição negativa do CLK.

Fig. 5.17 - Formas de Onda Ilustrativas do funcionamento do FLIP-FLOP JK.

5.5 - FLIP-FLOPS D e T

Além dos FLIP-FLOPS RS e JK, discutidos anteriormente, há dois outros tipos que são de uso mais restrito.

O FLIP-FLOP D possui uma única entrada, denominada DATA e sua tabela da verdade está mostrada na figura 5.18:

D	Qn+1
Ø	Ø
1	1

Fig. 5.18 - Tabela da Verdade para o FLIP-FLOP D

Pela tabela vemos que no FLIP-FLOP D a saída Q assume o estado da entrada como conseqüência do pulso do CLOCK.

O FLIP-FLOP T (TOGGLE)comporta-se de acordo com a tabela da verdade apresentada na figura 5.19:

T	Qn+1
Ø	Qn
1	$\overline{Q}$n

Fig. 5.19 - Tabela da Verdade para o FLIP-FLOP T.

Quando a entrada T está em "Ø", o FLIP-FLOP não muda de estado com a aplicação do pulso de CLOCK. Por outro lado, se a entrada T estiver em "1", o FLIP-FLOP passará para o estado oposto ao que se encontra antes da aplicação do CLOCK. Este tipo de FLIP-FLOP não é, em geral, encontrado como tal, sendo construído a partir de outros tipos. A figura 5.20 mostra as formas de onda para um FLIP-FLOP T gatilhado na transição negativa do CLOCK, admitindo que a entrada T seja mantida em "1" durante todo o tempo:

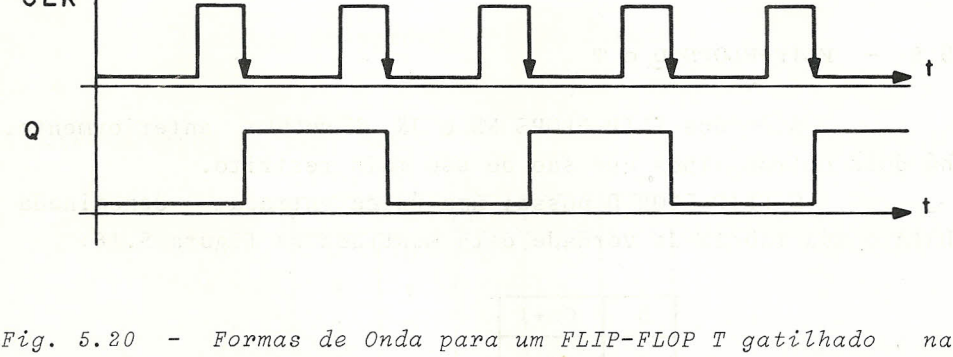
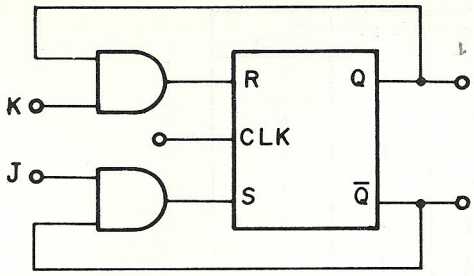


Fig. 5.20 - Formas de Onda para um FLIP-FLOP T gatilhado na descida do CLOCK (T = "1").

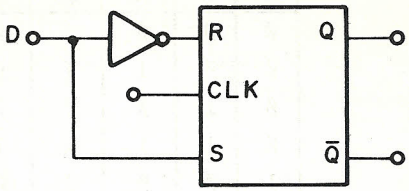
Observe que a freqüência do CLOCK aparece dividida por dois na saída, ou seja, a cada dois ciclos do CLOCK corresponde um ciclo da forma de onda da saída Q.

5.6 - Obtenção de um Tipo de FF a partir de outro

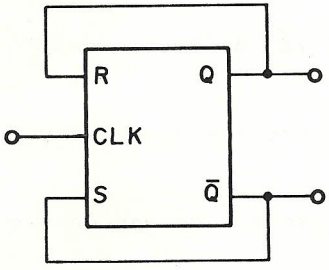
Dispondo de um FLIP-FLOP RS e algumas portas lógicas, podemos construir FLIP-FLOPS J-K, D e T, conforme ilustrado na figura 5.21:



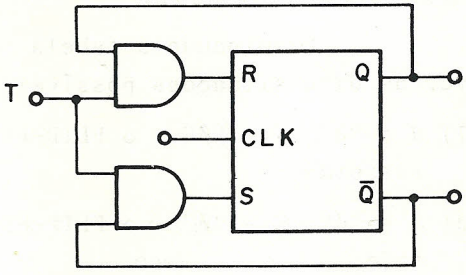
(a) R-S para J-K



(b) R - S para D



(c) R-S para T
(com T="1" sempre)



(d) R - S para T
(com entrada T)

Fig. 5.21 - FLIP-FLOPS J-K, D e T construídos a partir do FF RS.

O comportamento do circuito da figura 5.21(a) pode ser compreendido facilmente examinando a tabela apresentada na figura 5.22, onde Qn e $\overline{Q}$n são as saídas do circuito antes da aplicação do CLOCK e Qn+1, $\overline{Q}$n+1 após a mesma. O mesmo raciocínio que será detalhado a seguir pode ser aplicado aos outros circuitos apresentados neste item, ficando a cargo do leitor fazê-lo.

J	K	Qn	$\overline{Qn}$	R	S	Qn+1	$\overline{Qn+1}$	
Ø	Ø	Ø	1	Ø	Ø	Ø	1	1
Ø	Ø	1	Ø	Ø	Ø	1	Ø	2
Ø	1	Ø	1	Ø	Ø	Ø	1	3
Ø	1	1	Ø	1	Ø	Ø	1	4
1	Ø	Ø	1	Ø	1	1	Ø	5
1	Ø	1	Ø	Ø	Ø	1	Ø	6
1	1	Ø	1	Ø	1	1	Ø	7
1	1	1	Ø	1	Ø	Ø	1	8

Fig. 5.22 - Tabela da Verdade para o circuito da fig. 5.21(a).

Examinando a tabela da figura 5.22, podemos descrever as oito situações possíveis:

1) J = "Ø", K = "Ø" - o FLIP-FLOP estava ressetado e permanece ressetado;

2) J = "Ø", K = "Ø" - o FLIP-FLOP estava setado e permanece setado,

logo, para J = "Ø" e K = "Ø" não muda de estado.

3) J = "Ø", K = "1" - o FLIP-FLOP estava ressetado e permanece ressetado;

4) J = "Ø", K = "1" - o FLIP-FLOP estava setado e é ressetado, logo, para J = "Ø" e K = "1" o FLIP-FLOP estará ressetado após o pulso de CLOCK;

5) J = "1", K = "Ø" - o FLIP-FLOP está ressetado e é setado;

6) J = "1", K = "Ø" - o FLIP-FLOP está setado e permanece setado,

logo, para J = "1" e K = "Ø" o FLIP-FLOP estará setado após o pulso de CLOCK;

7) J = "1", K = "1" - o FLIP-FLOP está ressetado e é setado;

8) J = "1", K = "1" - o FLIP-FLOP está setado e é ressetado, logo, para J = "1" e K = "1" o FLIP-FLOP mudará para o estado oposto após o CLOCK.

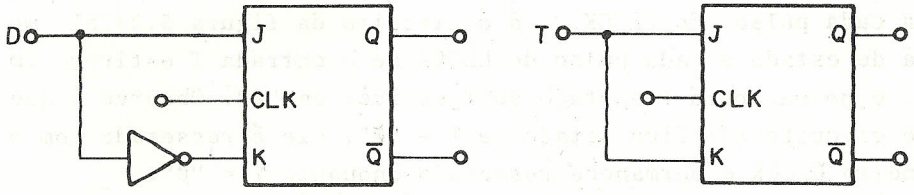
Concluímos da discussão acima que realmente o circuito da figura 5.21(a) se comporta como um FLIP-FLOP JK.

No circuito da figura 5.21(b), vemos que a entrada D= "Ø" fará com que o FLIP-FLOP seja ressetado e D = "1" fará com que ele seja setado.

O circuito da figura 5.21(c) corresponde a um FLIP-FLOP T com a entrada mantida sempre em "1", ou seja, ele mudará de estado a cada pulso de CLOCK.

Finalmente, o circuito da figura 5.21(d) corresponde a um FLIP-FLOP T com a entrada T disponível, se T = "Ø" ele não muda de estado e se T = "1" ele se complementa ou inverte.

Com o FLIP-FLOP J-K podemos construir os FLIP-FLOPS D e T conforme ilustra a figura 5.23:

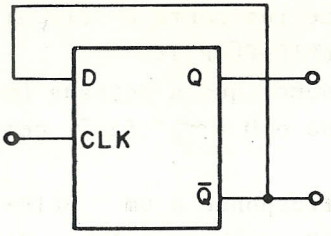


(a) J-K para D (b) J-K para T

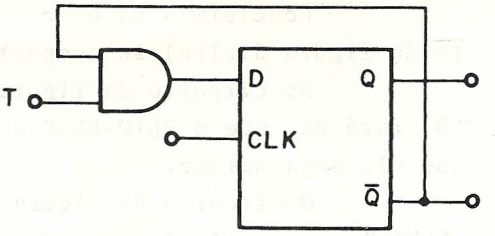
Fig. 5.23 - FLIP-FLOP D e T a partir do FLIP-FLOP J-K.

No circuito da figura 5.23(a), se D = "Ø" o FLIP-FLOP será ressetado e se D = "1" será setado. No circuito da figura 5.23(b), se T = "Ø" o FLIP-FLOP não muda de estado e, se T="1", ele se complementa.

Com o FLIP-FLOP D, podemos construir o FLIP-FLOP T de duas maneiras, conforme ilustra a figura 5.24.



(a) D para T
(com T="1", sempre)



(b) D para T
(com entrada T)

Fig. 5.24 - FLIP-FLOP T a partir do FLIP-FLOP D.

O circuito da figura 5.24(a) equivale a um FLIP-FLOP T no qual a entrada T esteja sempre em "1", pois mudará de estado a cada pulso de CLOCK. Já o circuito da figura 5.24(b) mudará de estado a cada pulso de CLOCK se a entrada T estiver em "1", e permanecerá ressetado se T estiver em "Ø". Observe que este circuito *não* fica setado se T = "Ø", ele é ressetado com o primeiro CLOCK e permanece ressetado enquanto T = "Ø".

5.7 - FLIP-FLOP da Família TTL

Há uma grande variedade de FLIP-FLOPS disponíveis na série 74TTL. Neste item, apresentaremos alguns deles, acompanhados de uma breve descrição. Os parâmetros indicados para cada FLIP-FLOP correspondem à versão standard e são valores típicos fornecidos pelos manuais.

Discutimos anteriormente o significado dos parâmetros t_H (HOLD TIME) e t_S (SET-UP TIME) para um FLIP-FLOP. Um outro dado muito importante é a freqüência máxima do CLOCK. Sabemos que um circuito não muda de estado instantaneamente; ele requer um certo tempo para que a mudança ocorra. No caso de um FLIP-FLOP, este tempo limita a freqüência do sinal de CLOCK que pode ser aplicado ao dispositivo. Se esta freqüência for excedida, o FLIP-FLOP não conseguirá mudar de estado entre dois pulsos de CLOCK sucessivos e o circuito funcionará de maneira im

previsível. A figura 5.25 mostra as freqüências máximas do sinal de CLOCK para as diversas versões de FLIP-FLOP da família TTL:

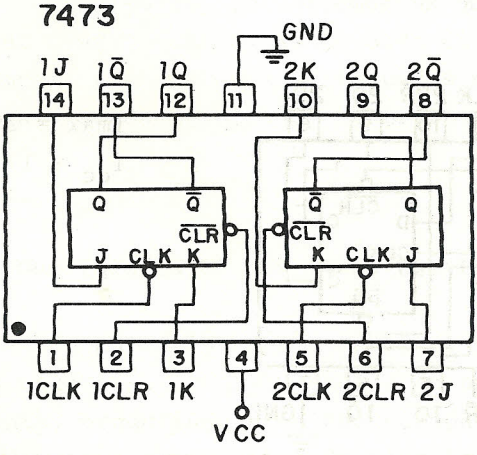
	74L	74	74LS	74H	74S
fmax (MHZ)	3	35	45	50	125

Fig. 5.25 - Freqüência máxima de CLOCK para FLIP-FLOPS da série 74.

Esta tabela serve como referência; para cada FLIP-FLOP individual deve-se consultar o manual do fabricante.

As entradas assíncronas de CLEAR e PRESET devem ser mantidas em um nível lógico definido. Deixá-las abertas, ou seja, flutuando, pode acarretar o gatilhamento espúrio do FLIP-FLOP. Estas entradas são sempre ativas no estado "Ø", de modo que se não forem usadas é comum conectá-las a V_{cc} através de um resistor.

7473 - DUAL J-K FLIP FLOP WITH CLEAR



fmax = 20 MHZ

I_{cc} = 20 mA/CI

Fig. 5.26 - 7473.

Contém dois FLIP-FLOPS J-K sensíveis ao nível do CLOCK (LEVEL TRIGGERED) com entrada de CLEAR assíncrono.

A tabela da verdade para este FLIP-FLOP está mostrada na figura 5.27:

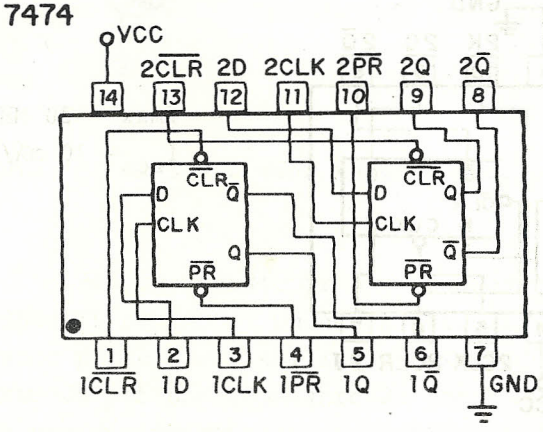
$\overline{CLR}$	CLK	J	K	Qn+1	$\overline{Q}$n+1
Ø	X	X	X	Ø	1
1	⎍	Ø	Ø	Qn	$\overline{Q}$n
1	⎍	Ø	1	Ø	1
1	⎍	1	Ø	1	Ø
1	⎍	1	1	$\overline{Q}$n	Qn

Fig. 5.27 - Tabela da Verdade para o CI 7473.

Nesta tabela, o símbolo " ⎍ " representa um pulso positivo na entrada de CLOCK.

As entradas J e K devem manter-se estáveis durante o pulso de CLOCK, caso contrário um gatilhamento errôneo do FLIP-FLOP poderá ocorrer.

7474 - DUAL D-TYPE POSITIVE-EDGE TRIGGERED FLIP-FLOP WITH PRESET AND CLEAR



fmax = 25 MHZ

I_{cc} = 17 mA/CI

Fig. 5.28 - 7474.

Contém dois FLIP-FLOPS do tipo D, gatilhados na transição positiva do CLOCK e com entradas de CLEAR e PRESET assíncronos. A tabela da verdade para este FLIP-FLOP está ilustrada na figura 5.29:

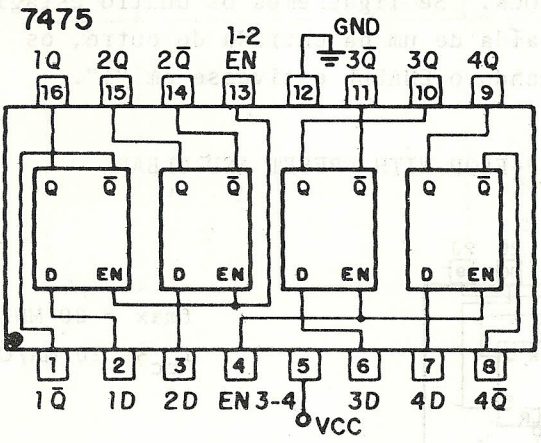
$\overline{CLR}$	$\overline{PR}$	D	CLK	Qn+1	$\overline{Q}$n+1
Ø	Ø	X	X	1	1
Ø	1	X	X	Ø	1
1	Ø	X	X	1	Ø
1	1	Ø	↑	Ø	1
1	1	1	↑	1	Ø

OBS: não permitida

Fig. 5.29 - Tabela da Verdade para o CI 7474.

Observe que as entradas $\overline{CLR}$ e $\overline{PR}$ não devem estar ativas simultaneamente, pois uma condição não permitida será assumida pelo FLIP-FLOP.

7475 - QUAD D-TYPE LATCH



t_{pd} = 24 ns
I_{cc} = 32 mA/CI

Fig. 5.30 - 7475.

Este CI contém quatro LATCHES do tipo D que são controlados aos pares pelas linhas de ENABLE (habilitação).

Quando a linha ENABLE está no nível "1", as saídas Q e $\overline{Q}$ acompanham a entrada D; o LATCH é transparente, ou seja, caso as entradas variem, as saídas variam também. Por outro lado, quando a linha ENABLE está no nível "Ø", as saídas não respondem às variações na entrada D; o LATCH armazena a informação que estava presente na entrada imediatamente antes da transição da linha ENABLE do nível "1" para o nível "Ø".

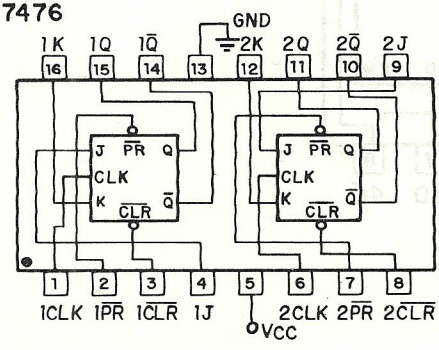
A tabela da verdade para este CI está ilustrada na figura 5.31:

EN	D	Qn+1	$\overline{Q}$n+1
Ø	X	Qn	$\overline{Q}$n
1	Ø	Ø	1
1	1	1	Ø

Fig. 5.31 - Tabela da Verdade para o CI 7475.

Observe que os diversos FLIP-FLOPS deste CI não se prestam a uma aplicação onde se quer apenas uma mudança de estado a cada pulso de CLOCK. Se ligássemos os quatro estágios em cascata, ou seja, a saída de um na entrada do outro, os quatro mudariam de estado quando o ENABLE estivesse em "1".

7476 - DUAL J-K FLIP-FLOP WITH PRESET AND CLEAR



fmax = 20 MHZ

I_{cc} = 20 mA/CI

Fig. 5.32 - 7476.

Contém dois FLIP-FLOPS J-K sensíveis ao nível do CLOCK (LEVEL TRIGGERED) com entradas de CLEAR e PRESET assíncrono.

A tabela da verdade para este FLIP-FLOP está ilustrada na figura 5.33:

$\overline{CLR}$	$\overline{PR}$	J	K	CLK	Qn+1	$\overline{Q}$n+1
Ø	Ø	X	X	X	1	1
Ø	1	X	X	X	Ø	1
1	Ø	X	X	X	1	Ø
1	1	Ø	Ø	⎍	Qn	$\overline{Q}$n
1	1	Ø	1	⎍	Ø	1
1	1	1	Ø	⎍	1	Ø
1	1	1	1	⎍	$\overline{Q}$n	Qn

OBS: não permitido

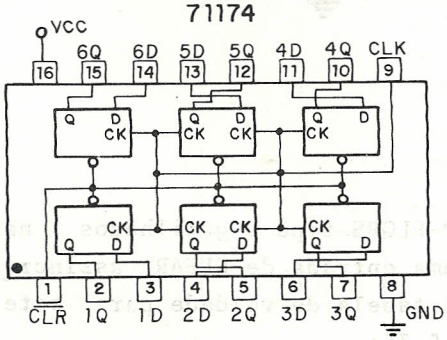
Fig. 5.33 - Tabela da Verdade para o CI 7476.

Nesta tabela, o símbolo ⎍ representa um pulso positivo na entrada de CLOCK.

As entradas J e K devem manter-se estáveis durante o pulso de CLOCK, caso contrário poderá ocorrer o gatilhamento errôneo do FLIP-FLOP.

As entradas $\overline{CLR}$ e $\overline{PR}$ nunca devem ficar ativas ao mesmo tempo, pois, neste caso, o FLIP-FLOP assumirá uma condição não permitida.

74174 - HEX D - TYPE FLIP-FLOP WITH CLEAR



tpd = 23 ns
fmax= 35 MHZ
I_{cc} = 45 mA/CI

Fig. 5.34 - 74174.

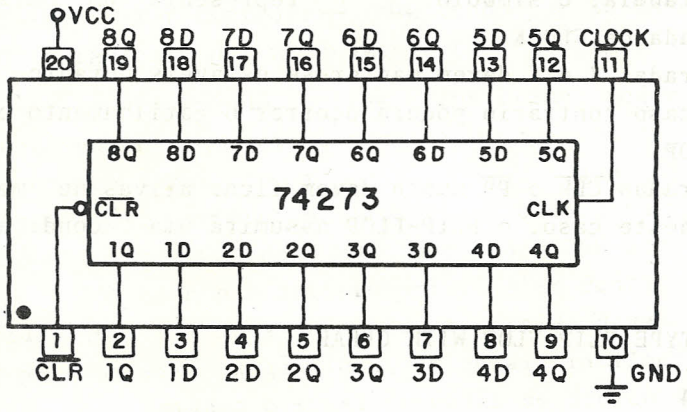
Contém 6 FLIP-FLOPS tipo D gatilhados na transição positiva do CLOCK, com uma entrada de CLEAR assíncrono, comum a todos os FLIP-FLOPS.

A tabela da verdade para este FLIP-FLOP está ilustrada na figura 5.35:

$\overline{CLR}$	D	CLK	Qn+1
Ø	X	X	Ø
1	Ø	↑	Ø
1	1	↑	1
1	X	Ø	Qn

Fig. 5.35 - Tabela da Verdade para os CI'S 74174 e 74273.

74273 - OCTAL D TYPE FLIP-FLOP WITH CLEAR

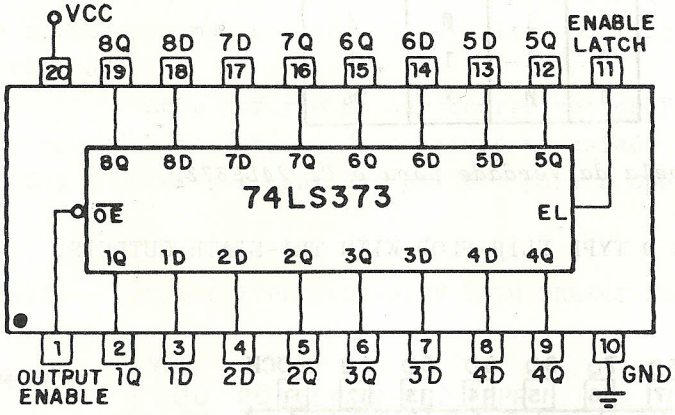


t_{pd} = 17 ns
fmax= 30 MHZ
I_{cc} = 62 mA/CI

Fig. 5.36 - 74273.

O 74273 contém oito FLIP-FLOPS tipo D gatilhados na transição positiva do CLOCK, com uma entrada de CLEAR assíncrono comum a todos os FLIP-FLOPS. A tabela da verdade para este FLIP-FLOP está mostrada na figura 5.35:

74LS373 - OCTAL D-TYPE TRANSPARENT LATCH



t_{pd} = 12 ns

fmax = 50 MHZ

I_{cc} = 24 mA/CI

Fig. 5.37 - 74LS373.

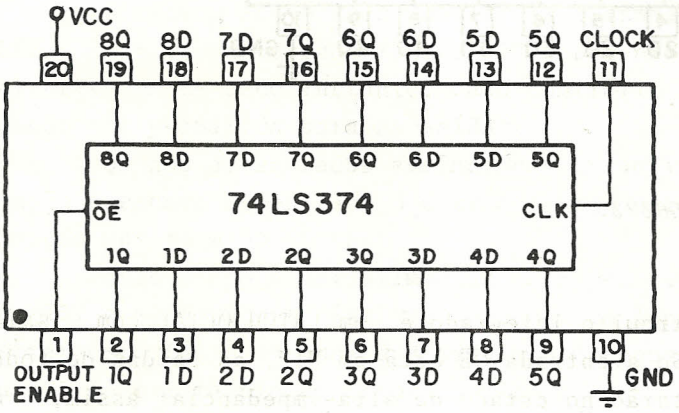
Este circuito integrado é um LATCH OCTAL com saída em TRI-ESTATE. Se a entrada $\overline{OE}$ está em "1", as saídas de todos os FLIP-FLOPS estarão no estado de alta-impedância; assim, as saídas de vários CI'S podem ser ligadas a um BUS comum desde que apenas um deles esteja habilitado em um determinado instante.

Se $\overline{OE}$ está ativo, igual a "Ø", o estado das saídas será controlado pela entrada EL. Se EL estiver em "1", dizemos que o LATCH está aberto; ele é transparente, pois o que estiver na entrada D aparecerá na saída Q correspondente e se a entrada D varia, a saída Q acompanha a variação. Se EL estiver em "Ø", a saída Q não acompanha as variações da entrada D correspondente, dizemos que o LATCH está fechado e a saída Q será o conteúdo da entrada D imediatamente antes da transição de EL de "1" para "Ø" . Isto significa que os FF'S são gatilhados na transição negativa da linha EL. A figura 5.38 ilustra a tabela da verdade para o CI:

$\overline{OE}$	EL	D	Qn+1
1	X	X	Hi-Z
Ø	1	Ø	Ø
Ø	1	1	1
Ø	Ø	X	Qn

Fig. 5.38 - Tabela da Verdade para o CI 74LS373.

74LS374 - OCTAL D TYPE FLIP-FLOP WITH TRI-STATE OUTPUTS.



tpd = 15 ns
fmax = 50 MHZ
I_{cc} = 27 mA/CI

Fig. 5.39 - 74LS374.

Este circuito integrado contém 8 FLIP-FLOPS tipo D gatilhados na transição positiva do CLOCK e com saída TRI-STATE . Se $\overline{OE}$ está em "1", todas as saídas estão no estado de alta impedância. A figura 5.40 mostra a tabela da verdade para este CI:

$\overline{OE}$	CLOCK	D	Qn+1
1	X	X	Hi-Z
Ø	↑	Ø	Ø
Ø	↑	1	1
Ø	L	X	Qn

Fig. 5.40 - Tabela da Verdade para o CI 74LS374.

5.8 - FLIP-FLOP da Família CMOS

Há diversos FLIP-FLOPS disponíveis na série 4ØØØ CMOS, e apresentaremos a seguir dois deles, que são os mais comumente utilizados.

Ao contrário do que ocorre com os FF'S da família TTL, as entradas assíncronas são ativas no estado lógico "1" (ACTIVE HIGH) e devem ser mantidas em "Ø" para a operação normal.

4Ø13 - DUAL D TYPE FLIP-FLOP WITH PRESET AND CLEAR

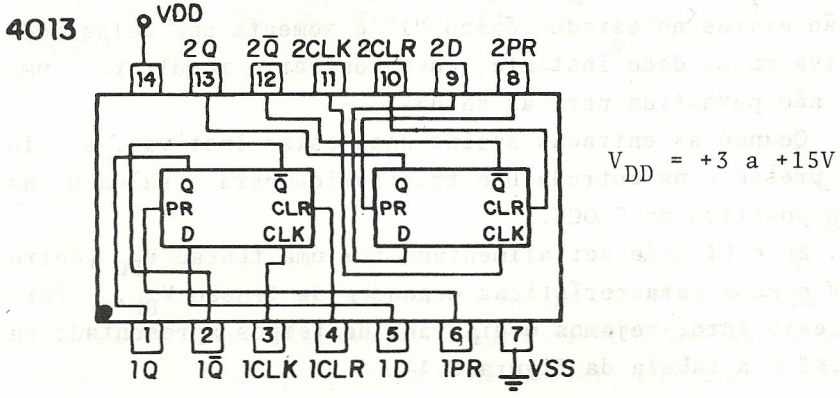


Fig. 5.41 - 4Ø13.

Contêm dois FLIP-FLOPS D gatilhados na transição positiva do CLOCK, com entradas de CLEAR e PRESET assíncrono. A tabela da verdade para cada FLIP-FLOP está ilustrada na figura 5.42:

PR	CLR	D	CLK	Qn+1	$\overline{Qn+1}$	OBS:
Ø	Ø	Ø	↑	Ø	1	
Ø	Ø	1	↑	1	Ø	
Ø	1	X	X	Ø	1	
1	Ø	X	X	1	Ø	
1	1	X	X	1	1	não permitido

Fig. 5.42 - Tabela da Verdade para o CI 4Ø13.

Conforme se vê pela tabela, as entradas de CLEAR e PRESET são ativas no estado lógico "1" e somente uma delas pode estar ativa em um dado instante, caso contrário resultará uma condição não permitida para as saídas.

Quando as entradas assíncronas estão inativas, a informação presente na entrada D é transferida para a saída Q na transição positiva do CLOCK.

Este CI pode ser alimentado com uma tensão V_{DD} entre +3 e +15V e suas características dependem da tensão V_{DD}. Para ilustrar este fato, vejamos o diagrama de tempos apresentado na figura 5.43 e a tabela da figura 5.44:

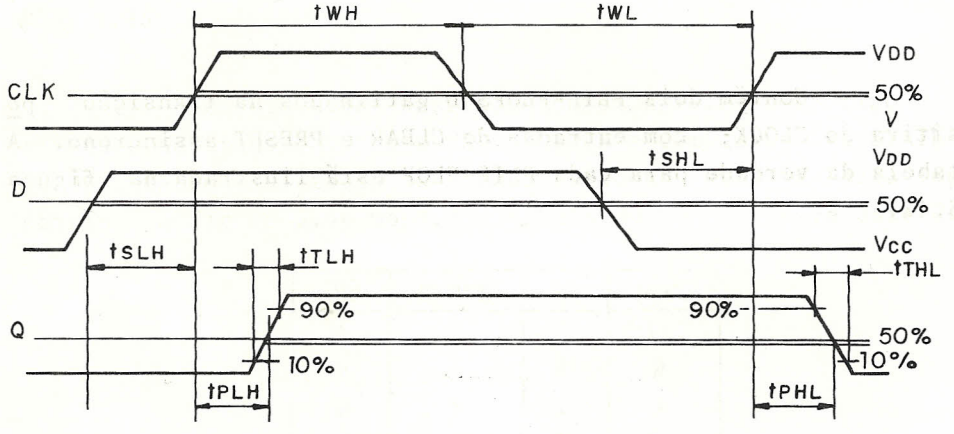


Fig. 5.43 - Formas de Onda para o FLIP-FLOP 4Ø13.

Na figura 5.43, os diversos tempos são medidos entre os níveis de 50% de V_{DD} das formas de onda, com exceção de t_{TLH} e t_{THL}, medidos entre os níveis de 10% e 90% da forma de onda da saída Q. Os tempos mostrados na figura 5.43 são assim definidos:

t_{WH} - largura mínima do pulso de CLOCK no estado lógico "1" (WIDTH HI);

t_{WL} - largura mínima do pulso de CLOCK no estado lógico "Ø" (WIDTH LO);

t_{SLH} - SET-UP TIME para a transição positiva da saída (LO TO HI);

t_{SHL} - SET-UP TIME para a transição negativa da saída(HI TO LO);

t_{pLH} - atraso da transição positiva da saída;

t_{pHL} - atraso da transição negativa da saída;

t_{TLH} - tempo de transição positiva ou tempo de subida;

t_{THL} - tempo de transição negativa ou tempo de descida.

Conforme indicamos anteriormente, estes parâmetros,bem como outros não apresentados na figura, dependem da tensão de alimentação. Veja na tabela abaixo os valores fornecidos pelo fabricante:

Símbolo	Mín.	Tip.	Máx.	Unidade	V_{DD}
t_{pLH}		15Ø	3ØØ	ns	5
t_{pHL}		75	11Ø	ns	1Ø
t_{TLH}		75	125	ns	5
t_{THL}		5Ø	7Ø	ns	1Ø
t_{WL}		125	2ØØ	ns	5
t_{WH}		5Ø	8Ø	ns	1Ø
t_{SLH}		2Ø	4Ø	ns	5
t_{SHL}		1Ø	2Ø	ns	1Ø
fmax	2,5	4,Ø		MHZ	5
	7,Ø	1Ø,Ø		MHZ	1Ø
t_{rCK}			15	µs	5
t_{FCK}			5	µs	1Ø

Fig. 5.44 - Características Dinâmicas do CI 4Ø13.

Na tabela acima, fmax é a freqüência máxima do CLOCK e t_{rCK} e t_{fCK} são os tempos de subida (RISE TIME) e descida (FALL TIME) do sinal de CLOCK.

Vamos interpretar a tabela e verificar que condições devem ser satisfeitas para que o CI funcione corretamente. Admitamos que a tensão de alimentação seja V_{DD} = 1Ø V. Neste caso, o fabricante nos diz o seguinte:

- a frequência máxima do CLOCK deverá ser 1Ø MHZ (valor típico) e garantidamente o CI funcionará com um CLOCK de 7 MHZ ou menos (valor mínimo de f max);
- o sinal de CLOCK deve ter um tempo de subida e de descida inferior a 5 µs, caso contrário não se pode garantir o gatilhamento correto do FLIP-FLOP;
- o CLOCK deverá permanecer no estado lógico "1" por 50 ns no mínimo (valor típico); permanecendo por 80 ns ou mais no nível "1" garante-se o funcionamento correto (t_{WH});
- o CLOCK deverá permanecer no estado lógico "Ø" por 50 ns no mínimo (valor típico); permanecendo por 80 ns ou mais no nível "Ø" garante-se o funcionamento correto (t_{WL}).

Resumindo os três itens acima, podemos dizer que o funcionamento correto é garantido para uma freqüência de CLOCK menor que 7 MHZ, com t_{WH} e t_{WL} maiores que 80 ns. Observe que é necessário atender às três exigências simultaneamente. Por exemplo, um CLOCK de 5 MHZ (período = 2ØØ ns) com t_{WH} = 20 ns e t_{WL} = 18Ø ns não é satisfatório. Além destes três pontos, é claro que os tempos de subida e descida devem ser inferiores a 5 µs, mas esta exigência é, em geral, facilmente satisfeita.

Estando o sinal de CLOCK de acordo com as especificações, o fabricante assegura que:

- se a entrada D estiver estável no mínimo 5Ø ns(típico) ou 7Ø ns (máximo) antes da transição positiva do CLOCK (SET-UP TIME), o gatilhamento ocorrerá de maneira correta e as saídas mudarão de estado 75 ns (típico) ou 11Ø ns (máximo) após a transição positiva do CLOCK:

Esta discussão nos mostra quão importante é a utilização e interpretação correta dos dados fornecidos no manual.

4Ø27 - DUAL JK FLIP-FLOP WITH PRESET AND CLEAR

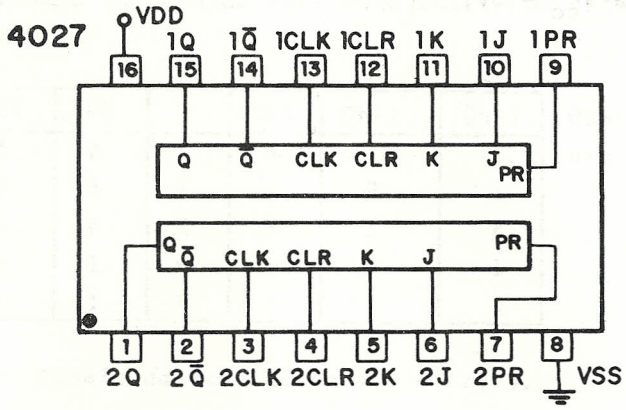


Fig. 5.45 - 4Ø27.

Este circuito integrado contém dois FLIP-FLOPS JK gatilhados na transição positiva do CLOCK, com entradas de CLEAR e PRESET assíncrono. A tabela da verdade para cada FLIP-FLOP está ilustrada na figura 5.46:

CLR	PR	J	K	CLK	Qn+1	$\overline{Q}$n+1
Ø	Ø	Ø	Ø	↑	Qn	$\overline{Q}$n
Ø	Ø	Ø	1	↑	Ø	1
Ø	Ø	1	Ø	↑	1	Ø
Ø	Ø	1	1	↑	$\overline{Q}$n	Qn
1	Ø	X	X	X	Ø	1
Ø	1	X	X	X	1	Ø
1	1	X	X	X	1	1

Fig. 5.46 - Tabela da Verdade para o CI 4Ø27.

Conforme se vê pela tabela, as entradas de CLEAR e PRESET assíncrono são ativas no estado "1" e somente uma delas pode estar ativa em um dado instante, caso contrário resultará

uma condição não permitida para as saídas.

A freqüência máxima do CLOCK é de 8 MHZ para V_{DD} = 10V e 3 MHZ para V_{cc} = 5V.

CAPÍTULO 6 - MULTIVIBRADORES ASTÁVEIS, MONOESTÁVEIS E DISPARADORES SCHMITT

6.1 - Introdução

No capítulo anterior, estudamos os multivibradores biestáveis ou FLIP-FLOPS e vimos que os mesmos apresentam dois estados estáveis, podendo ficar indefinidamente em um deles. Observamos também que, na maioria das aplicações, desejamos que as mudanças de estado de um circuito seqüencial ocorram em sincronismo com um pulso de CLOCK. Não nos preocupamos com a maneira de gerar este sinal de CLOCK, admitindo que havia algum circuito que o produzisse. Neste capítulo, discutiremos diversos circuitos que podem ser utilizados com esta finalidade.

Após estudarmos os geradores de CLOCK, muitos dos quais utilizam multivibradores astáveis, que são circuitos sem nenhum estado estável, veremos um terceiro tipo de multivibrador. Trata-se do multivibrador monoestável, que é um circuito com um estado estável, utilizado para gerar um pulso de largura conhecida, como resposta a um sinal de controle externo.

No estudo dos multivibradores astáveis e monoestáveis, apresentaremos um circuito, integrado muito usado na prática, que é o temporizador 555, descrevendo suas características e mostrando como utilizá-lo.

Finalmente, estudaremos um circuito muito útil na conversão de sinais com formas de onda diferente de pulsos (senoides, rampas, etc) em pulsos adequados para uso em sistemas digitais; trata-se do disparador Schmitt.

6.2 - Osciladores RC com Circuitos Integrados

Um multivibrador astável é um circuito que não possui estados estáveis, ou seja, oscila de um estado para outro constantemente, produzindo um sinal periódico cuja freqüência é determinada pelos componentes do circuito. Apenas como ilustração, a figura 6.1 mostra o esquema de um multivibrador astável

construído com componentes discretos.

Para compreender o funcionamento deste circuito, admitamos que T_1 esteja cortado (OFF) e T_2 saturado (ON). Neste caso, C_1 se carregará com uma tensão aproximadamente igual a V_{cc} e a polaridade mostrada na figura. Quando T_1 conduz novamente, aterrando o terminal esquerdo de C_1 a tensão de base de T_2 fica negativa (aproximadamente $-V_{cc}$) e T_2 vai para o corte. Enquanto T_2 está OFF, C_2 se carrega com uma tensão aproximadamente igual a V_{cc} de modo que seu terminal esquerdo fica negativo em relação ao direito. Enquanto isso, C_1 vai se descarregando e quando sua descarga está completa T_2 volta a conduzir. Neste instante, a base de T_1 recebe a tensão negativa causada pela carga de C_2 e T_1 vai para o corte. O processo se repete indefinida e periodicamente enquanto a alimentação estiver aplicada ao circuito.

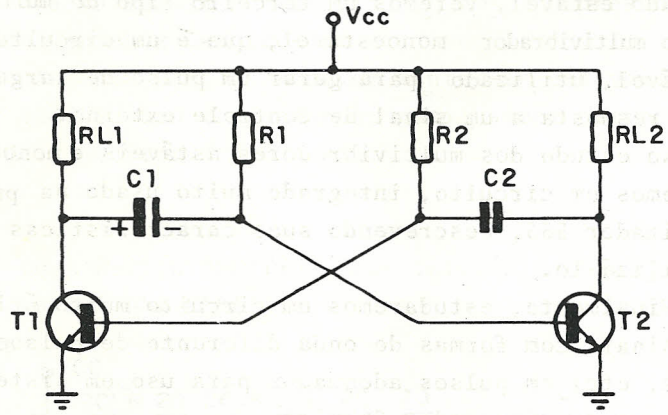


Fig. 6.1 - Multivibrador Astável acoplado por coletor.

Podemos construir osciladores utilizando circuitos integrados TTL ou CMOS. Quando a estabilidade de freqüência não é o fator primordial, utilizamos osciladores cuja freqüência é determinada por uma constante de tempo RC. Se a estabilidade for de grande importância, preferimos osciladores a cristal.

Vejamos, em primeiro lugar, alguns osciladores que utilizam circuitos integrados CMOS.

A figura 6.2 ilustra o circuito básico de um astável com inversores CMOS:

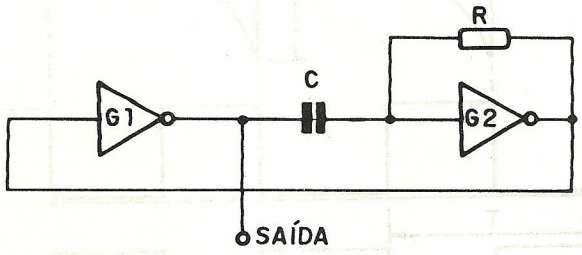


Fig. 6.2 - Astável com inversores CMOS.

Vamos admitir que a saída de G_2 esteja em "1". A saída de G_1 estará em "Ø" e o capacitor C carrega-se através de R. Quando a tensão sobre C atinge o limiar que G_2 entende como nível "1" e que é aproximadamente $\frac{1}{2}$ de V_{CC}, a saída de G_2 vai para "Ø" e a saída de G_1 para "1". Como a tensão no capacitor não pode alterar-se instantaneamente, a entrada de G_2 recebe um acréscimo de valor V_{CC}, realimentando ainda mais a mudança de estado de G_2. Agora o capacitor se carrega no sentido contrário e, quando a tensão na entrada de G_2 volta para $V_{CC}/2$, a saída desta porta vai novamente para "1". Em conseqüência, a saída de G_1 vai para "Ø", a variação de $-V_{CC}$ é realimentada para a entrada de G_2 e o processo se repete indefinidamente. A figura 6.3 ilustra as formas de onda para o circuito da figura 6.2:

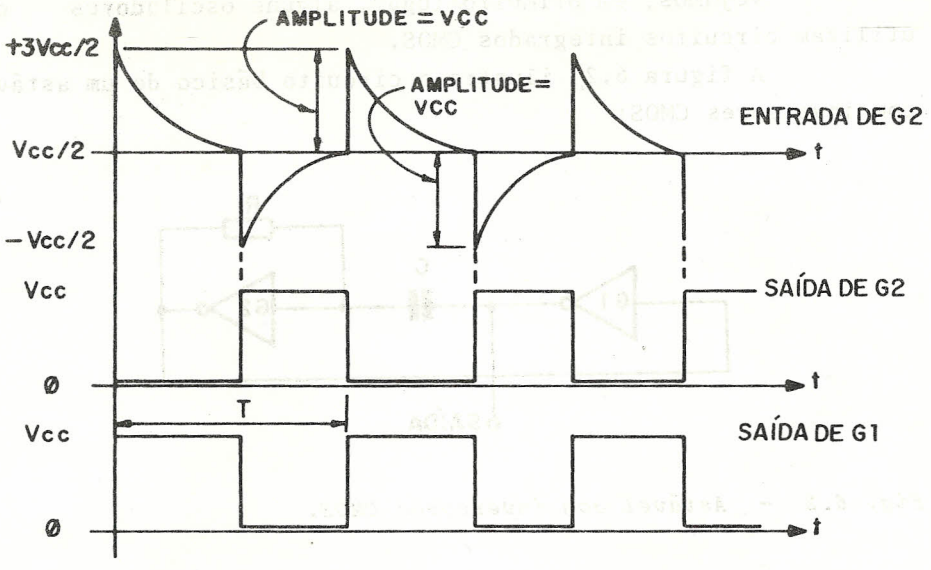


Fig. 6.3 - Formas de onda para o oscilador da figura 6.2.

A freqüência de oscilação deste circuito é dada pela expressão aproximada:

$$T = 2,2.R.C$$

$$f = \frac{1}{2,2RC}$$

Normalmente, recomenda-se a utilização de circuitos da série B pois os mesmos permitem conseguir menores tempos de comutação.

Conforme vimos anteriormente, os circuitos integrados CMOS possuem diodos de proteção na entrada para eliminar o problema de acúmulo de cargas estáticas. Estes diodos de proteção podem influir desfavoravelmente no comportamento do oscilador da figura 6.2 e, para contornar este problema, é recomendável o uso de um resistor de isolação R_I como ilustra a figura 6.4:

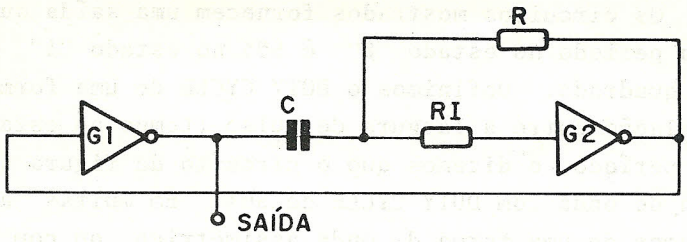
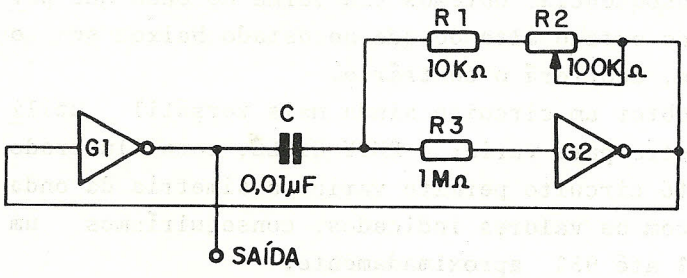


Fig. 6.4 - Oscilador CMOS com Resistor de Isolação R_I.

O valor de R_I deve ser no mínimo 1Ø vezes maior que o de R e seu limite máximo é determinado pelo tempo que se pode permitir para a carga da capacitância de entrada de G_2, que é da ordem de 5 pF. Normalmente o valor de R_I é de algumas centenas de quilo ohms. O período de oscilação T continua aproximadamente igual a 2,2R.C, desde que R_I > 1Ø R.

O oscilador da figura 6.4 pode ser utilizado na geração de sinais de freqüência desde décimos de Hertz até centenas de KHZ. O valor de C não deve ser inferior a 50 pF e R pode variar desde alguns KΩ até alguns MΩ.

Se o resistor R for constituído de uma parte fixa e uma variável, teremos um oscilador de freqüência variável, conforme ilustra a figura 6.5. Neste capítulo, C determina a faixa de operação, R_1 a freqüência máxima de operação e $R_1 + R_2$ a freqüência mínima; neste caso, a freqüência pode ser variada sobre uma relação de mais de 1Ø para 1.

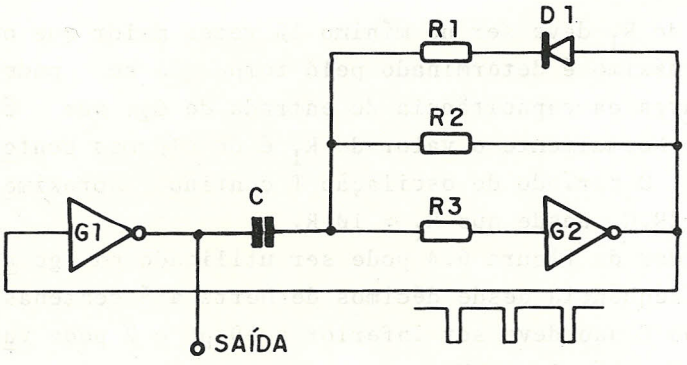


$$fmin = \frac{1}{2,2(R_1+R_2)C} = \simeq 413HZ$$

$$fmax = \frac{1}{2,2R_1C} = 4,55\ KHZ$$

Fig. 6.5 - Oscilador CMOS de freqüência variável.

Os circuitos mostrados fornecem uma saída que permanece 50% do período no estado "Ø" é 50% no estado "1", ou seja , uma onda quadrada. Definimos o DUTY CYCLE de uma forma de onda como a relação entre a largura de pulso (tempo no estado lógico "1") e o período, e dizemos que o circuito da figura 6.4 gera uma forma de onda com DUTY CYCLE de 50%. Em muitas aplicações necessitamos de uma forma de onda assimétrica, ou com DUTY CYCLE diferente de 50%. A solução é fazer com que o capacitor C tenha caminhos diferentes para a carga em um e outro sentido, conforme ilustra a figura 6.6:



$t_{LO}=1,1(R_1 \;//\; R_2).C_1$

$t_{HI}=1,1\ R_2.C_1$

$f = \dfrac{1}{t_{LO} + t_{HI}}$

Fig. 6.6 - Oscilador CMOS com saída Assimétrica.

Neste circuito a carga do capacitor de $3V_{cc}/2$ até $V_{cc}/2$ se faz atraves de R_2; a carga de $-V_{cc}/2$ até $V_{cc}/2$ se faz por R_1 em paralelo com R_2, pois, neste caso, o diodo D_1 está conduzindo. Em conseqüência, obtemos uma forma de onda que permanece mais tempo no estado alto do que no estado baixo se o diodo for invertido, ocorrerá o contrário.

Podemos obter um circuito ainda mais versátil utilizando um potenciômetro para variar o DUTY CYCLE, como ilustrado na figura 6.7. Este circuito permite variar a simetria da onda numa larga faixa, com os valores indicados, conseguiríamos um DUTY CYCLE desde 5% até 95% aproximadamente.

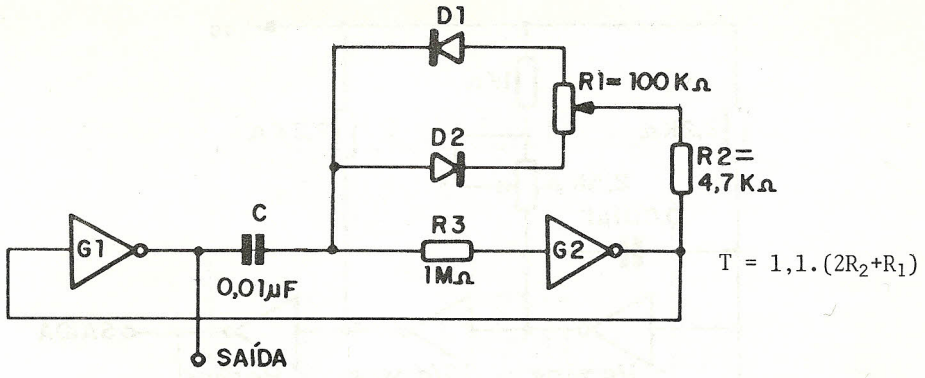


Fig. 6.7 - Oscilador CMOS com DUTY CYCLE variável entre 5% e 95%.

Um oscilador cuja oscilação pode ser interrompida por um sinal externo está ilustrado na figura 6.8. Neste circuito, se a linha de controle $\overline{\text{ENABLE}}$ está em "1", a saída permanece em "1", quando $\overline{\text{ENABLE}}$ vai para "Ø" a oscilação se inicia.

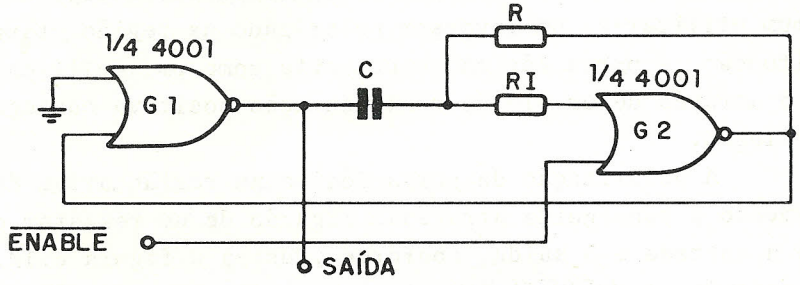


Fig. 6.8 - Oscilador CMOS com entrada de controle.

Um oscilador utilizando inversores TTL com saída em coletor aberto é ilustrado na figura 6.9. Este oscilador é capaz de gerar uma onda quadrada com freqüência variável entre 1MHZ e 1ØMHZ, mas sua estabilidade não é muito grande.

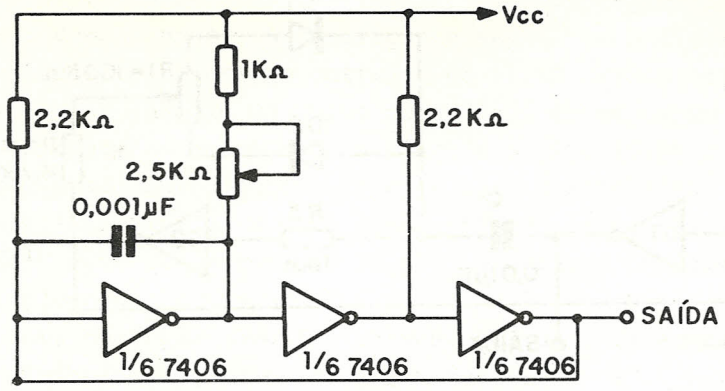


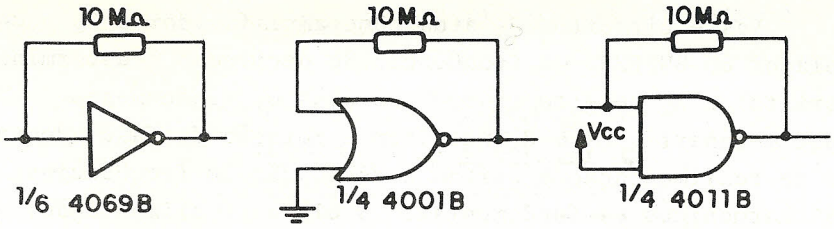
Fig. 6.9 - Oscilador TTL utilizando inversores com saída em Coletor Aberto.

6.3 - Osciladores a Cristal

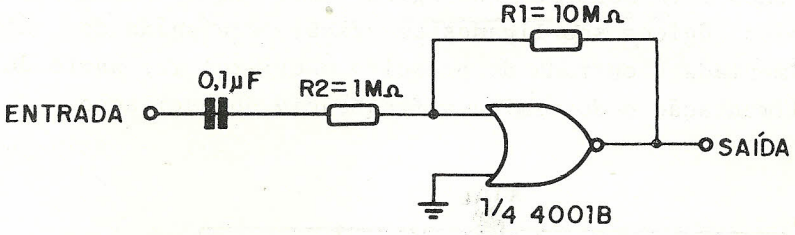
Quando se requer uma grande estabilidade de freqüência, a solução mais comum é o uso de um oscilador a cristal.

Num oscilador a cristal com circuitos integrados CMOS é comum utilizar-se um inversor polarizado na região ativa: nesta situação, a porta lógica se comporta como um amplificador linear e através de um elo de realimentação positiva consegue - se a oscilação.

A polarização da porta lógica na região ativa de funcionamento é conseguida através da ligação de um resistor de 10MΩ entre a entrada e a saída, conforme ilustra a figura 6.10. Com este arranjo, a saída ficará num valor quiescente de $V_{cc}/2$ e a porta lógica se comportará como um amplificador linear de alto ganho, com uma excursão de saída de $V_{cc}/2$ para mais ou para menos. A figura 6.10 ilustra também a aplicação de uma porta lógica como amplificador de ganho 10; o ganho é, neste caso, determinado pela relação R_1/R_2, à maneira de um amplificador operacional.



(a) Polarização



(b) Amplificador Linear de ganho 1Ø

Fig. 6.10 - Portas Lógicas CMOS Polarizadas na Região Ativa.

Neste tipo de aplicação, recomenda-se o uso de circuitos integrados da série B, operados com tensões de alimentação superiores a 4,5 V.

A figura 6.11 ilustra o circuito de um oscilador a cristal, utilizando uma porta lógica como amplificador:

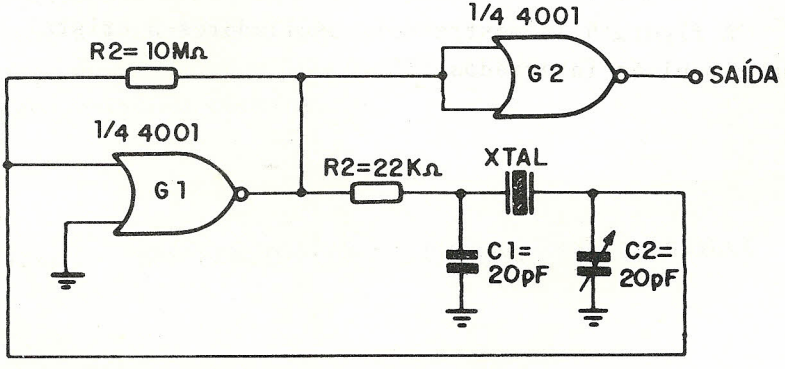


Fig. 6.11 - Osciladores a Cristal com CMOS.

Neste circuito, G_1 atua como amplificador e G_2 como um isolador ou BUFFER. A freqüência de oscilação é determinada pelo cristal e C_2 permite um pequeno ajuste, tipicamente de 1KHZ por megahertz R_1 e C_1 permitem a rotação de fase adequada no elo de realimentação e evitam a oscilação em freqüências no 3º e 5º harmônicos da fundamental. O elo de realimentação, que inclui o cristal, é conhecido como rede PI. A saída é capaz de alimentar uma entrada TTL.

Um outro oscilador a XTAL utilizando circuitos integrados CMOS está mostrado na figura 6.12. Neste circuito, três inversores lógicos são ligados em cascata e a saída do último é realimentada à entrada do primeiro o cristal faz parte do elo de realimentação e determina a freqüência de oscilação.

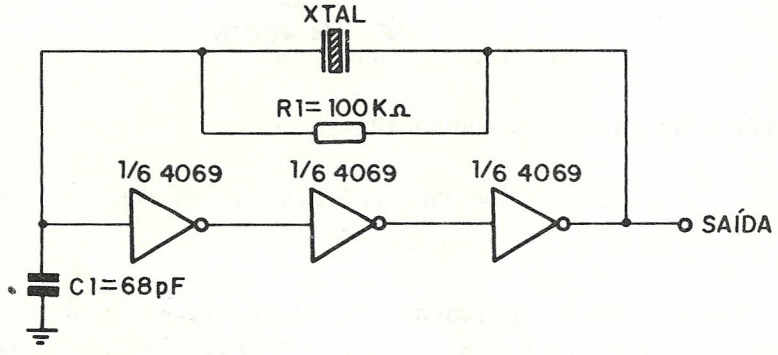


Fig. 6.12 - Oscilador a Cristal com CMOS.

A figura 6.13 mostra dois osciladores a cristal que utilizam circuitos integrados TTL:

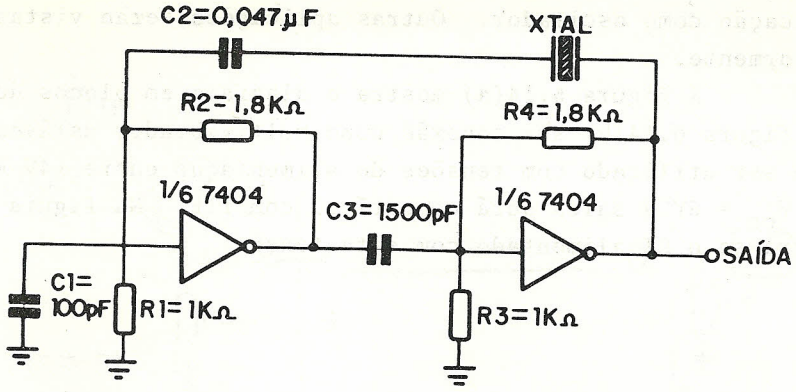


Fig. 6.13.a

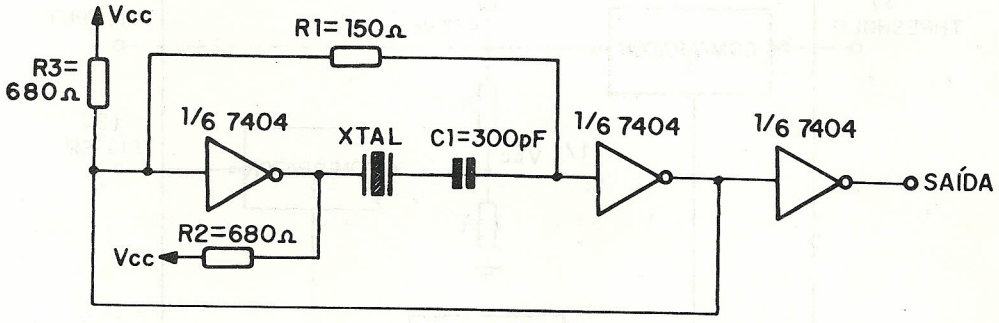


Fig. 6.13.b

Fig. 6.13 - Osciladores a Cristal com TTL.

6.4 - Temporizador 555

Um circuito integrado extremamente versátil e que encontra extensa aplicação prática é o 555 ou sua versão DUAL 556. Vários fabricantes produzem este temporizador (TIMER), que pode ser usado como multivibrador astável ou monoestável, gerador de ondas quadradas, gerador de pulsos, etc. A importância deste CI justifica que dediquemos um item a seu estudo e vejamos sua

aplicação como oscilador. Outras aplicações serão vistas posteriormente.

A figura 6.14(a) mostra o diagrama em blocos do 555 e a figura 6.14(b) sua conexão como multivibrador astável. O 555 pode ser utilizado com tensões de alimentação entre +4V e +15V; com V_{cc} = 5V a saída será compatível com TTL. Na figura 6.14(b) mostra-se o CI alimentado com esta tensão:

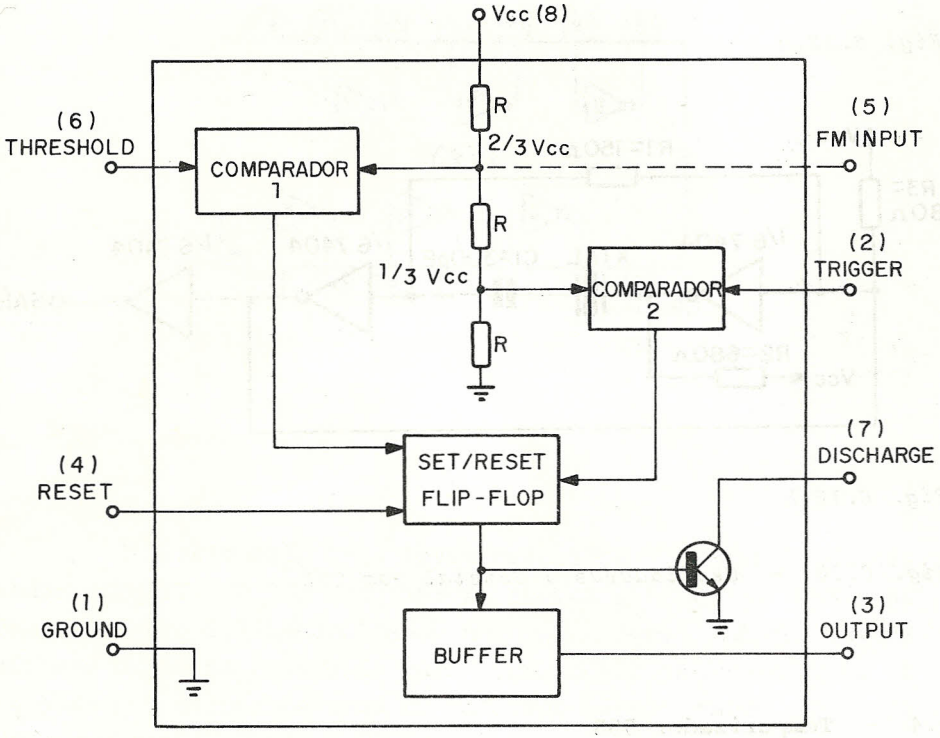


Fig. 6.14 (a) Diagrama em Blocos do 555.

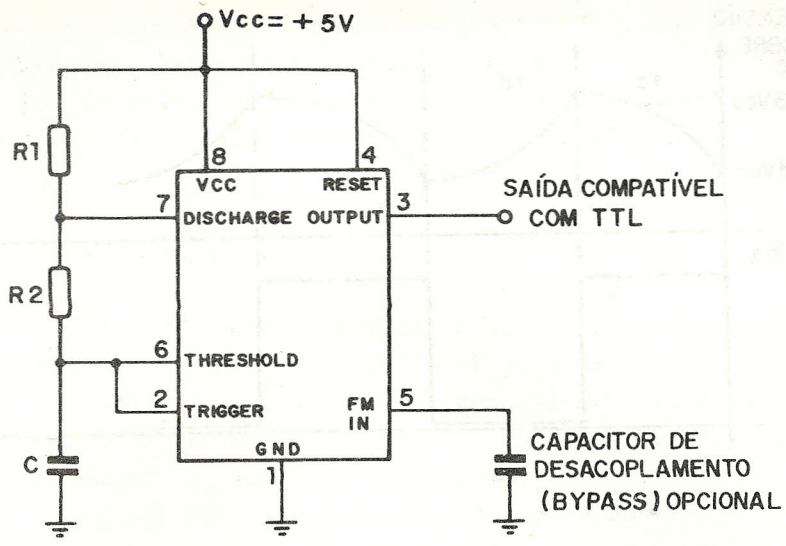
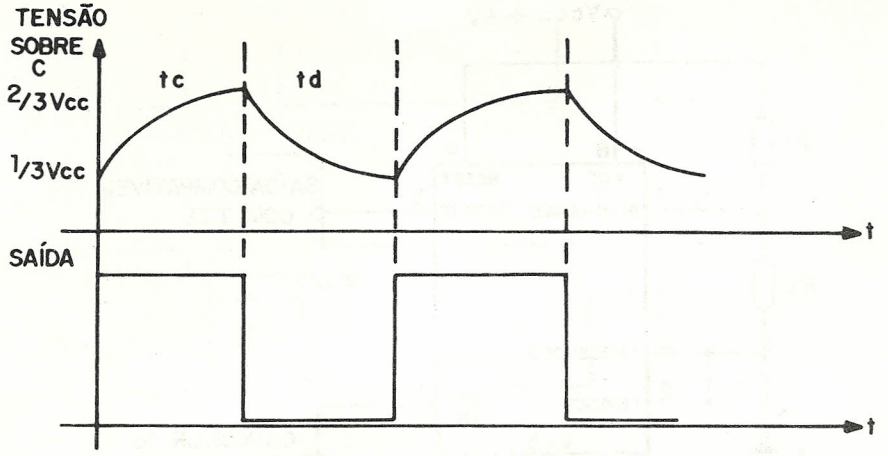


Fig. 6.14 (b) Ligação do 555 como astável.

Fig. 6.14 - Temporizador 555.

Para compreender o funcionamento do astável, vamos supor que, inicialmente, C esteja descarregado; a saída estará em "1" e o transistor de descarga estará cortado. O capacitor se carrega através de R_1 e R_2 e quando a tensão sobre ele atinge $^2/_3$ de V_{cc} o comparador (1) seta o FLIP-FLOP de controle. Em consequência, o transistor de descarga conduz; a saída passa para o estado lógico "Ø". O capacitor se descarrega através de R_2 e do transistor e, quando sua tensão atinge o valor $^1/_3$ de V_{cc}, o comparador (2) resseta o FLIP-FLOP de controle. A saída vai para o estado "1" novamente, o transistor deixa de conduzir e o capacitor se carrega novamente, repetindo o ciclo. As formas de onda correspondentes e as equações de projeto estão mostradas na figura 6.15:



t_c ⟹	tempo de carga	$t_c = 0,693.(R_1+R_2).C$
t_d ⟹	tempo de descarga	$t_d = 0,693 . R_2C$
T ⟹	período de oscilação	$T = t_c + t_d = 0,693.(R_1+ + 2R_2).C$
f ⟹	freqüência de oscilação	$f = 1/T = 1,44/(R_1+2R_2).C$

Fig. 6.15 - Formas de Onda e Equações de Projeto para o Astável da figura 6.14(b).

O valor máximo para $R_1 + R_2$ é de 3,3 MΩ e o valor mínimo para R_1 ou R_2 é de 1 KΩ, o que permite uma faixa de ajuste de freqüência da ordem de 3300: 1 em potencial. O valor mínimo do capacitor C é de 1000 pF e a freqüência máxima de oscilação 1 MHZ, o melhor desempenho é conseguido para freqüências menores que 300KHZ. No circuito da figura 6.14(b), o tempo em que a saída permanece em "1" é sempre maior que o tempo em que ela permanece em "0", pois o capacitor se carrega por R_1 e R_2 e descarrega apenas através de R_2. Assim, o DUTY CYCLE é sempre maior que 50%. No entanto, se R_2 for muito maior que R_1, por exem

plo R_2 = 1 MΩ e R_1 = 1 KΩ, teremos na saída uma onda quadrada quase perfeita. No circuito da figura 6.14(b), podemos conseguir uma saída com o DUTY CYCLE desde praticamente 50% até 99,9%, através da seleção de R_2 e R_1. Quanto maior a relação R_2/R_1 tanto mais o DUTY CYCLE se aproximará de 50%; inversamente, quanto menor a relação R_2/R_1 tanto mais o DUTY CYCLE se aproximará de 100%.

Se desejarmos um DUTY CYCLE menor que 50%, será necessário introduzir uma modificação no circuito, de modo que o capacitor se carregue através de uma resistência menor que a de descarga. Um tal circuito é mostrado na figura 6.16, onde a carga se faz através de R_1 apenas e a descarga através de R_2 apenas, permitindo assim obter qualquer DUTY CYCLE praticamente.

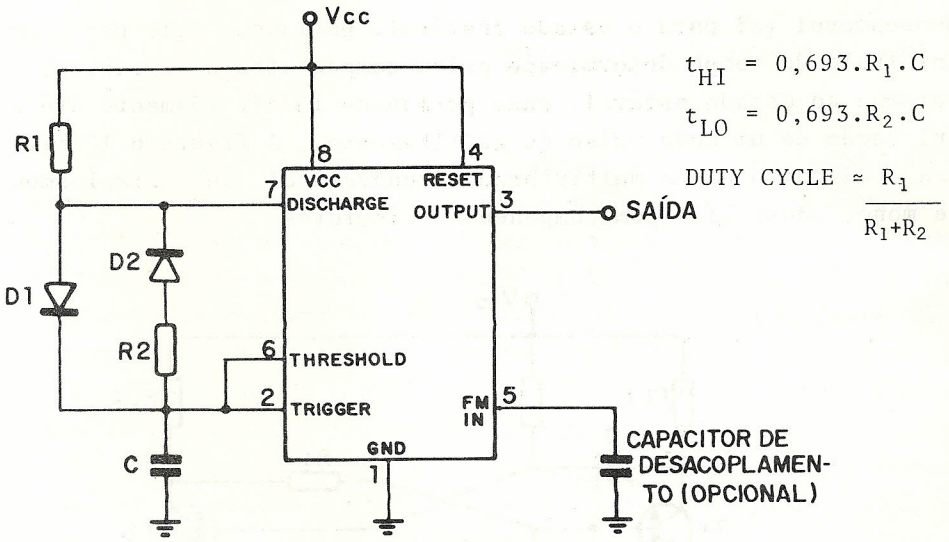


Fig. 6.16 - Astável com DUTY CYCLE variável entre ≈ 0% e 100%.

Nos circuitos mostrados nas figuras 6.14(b) e 6.16, a entrada RESET está ligada a V_{cc}. Este é o procedimento normal em circuitos astáveis; em outras aplicações a entrada RESET po

de ser usada para controle, uma vez que um nível lógico "Ø"aplicado a ela fará com que a saída vá para "Ø".

A entrada FM IN pode ser utilizada para modular em freqüência o sinal de saída, através da aplicação de um sinal AC nesta entrada.

O 555 é ideal para aplicações onde a estabilidade de freqüência requerida seja maior que 1%. Com ele é possível conseguir pulsos de período desde alguns microsegundos até horas . Para tempos muito grandes o valor de C é limitado pela corrente de fuga.

6.5 - Multivibradores Monoestáveis

O multivibrador monoestável é um circuito que possui apenas um estado estável. Através de um sinal de controle, o monoestável vai para o estado instável, permanece nele por um intervalo de tempo determinado pelos componentes do circuito, e retorna ao estado estável, onde permanece indefinidamente até a aplicação de um novo pulso de gatilhamento. A figura 6.17 ilustra o circuito de um multivibrador monoestável, ou simplesmente mono, construído com componentes discretos:

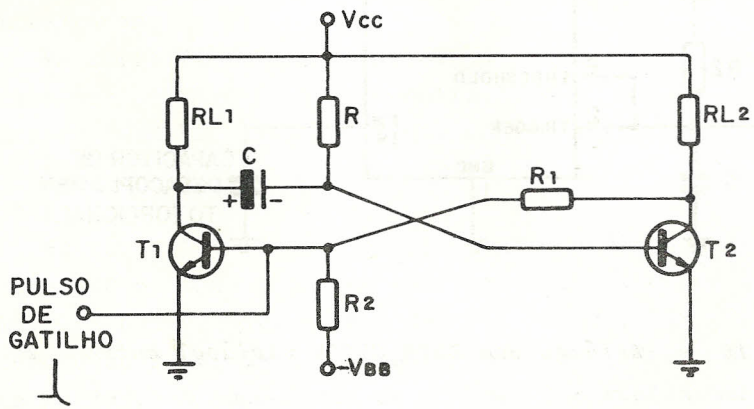
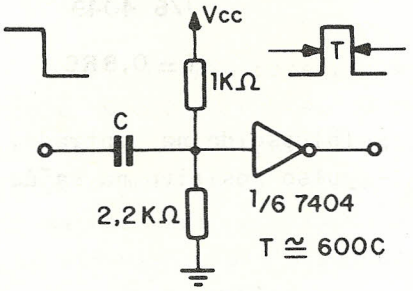


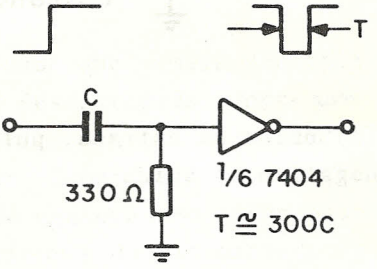
Fig. 6.17 - Multivibrador Monoestável.

Vamos admitir que T_1 esteja OFF e T_2 ON. O capacitor C irá se carregar com uma tensão aproximadamente igual a V_{cc} e a polaridade indicada. Se aplicarmos um estreito pulso de gatilho à base de T_1, de modo a fazê-lo conduzir, o terminal esquerdo de C será aterrado, aplicando um pulso negativo à base de T_2 e levando-o ao corte. Com T_1 ON e T_2 OFF, o capacitor C irá se descarregando através de R e, quando a tensão na base de T_2 estiver suficientemente positiva, este transistor conduzirá. A condição de T_2 levará T_1 ao corte e o circuito voltará ao estado inicial.

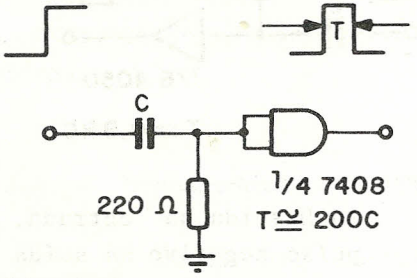
Antes de passarmos ao estudo dos monoestáveis com circuitos integrados, é conveniente examinarmos uma classe de circuitos denominados detetor de borda (EDGE DETECTOR) ou meio-monoestável. Estes circuitos diferem do monoestável propriamente dito no sentido de que o pulso de gatilhamento deve ter necessariamente uma duração maior que a do pulso de saída. A figura 6.18 ilustra alguns destes circuitos com CI'S da família TTL, mostrando as formas de onda e as durações do pulso obtidas na saída:



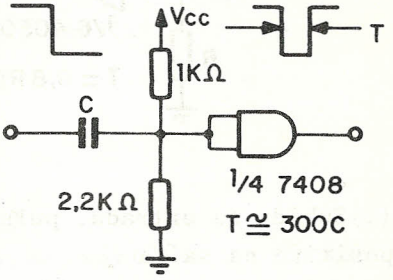
(a) Descida na entrada, pulso positivo na saída



(b) Subida na entrada, pulso negativo na saída



(d) Subida na entrada, pulso positivo na saída

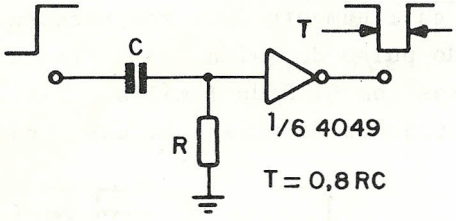


(d) Descida na entrada, pulso negativo na saída

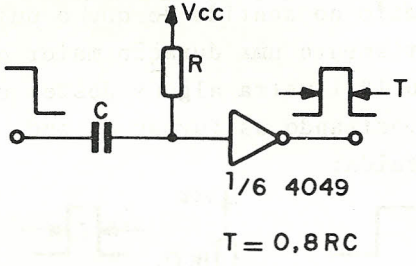
Fig. 6.18 - Detetores de Borda com TTL.

Em todos os casos, o circuito se baseia na diferenciação do sinal de entrada, é indispensável que este permaneça estável por um tempo maior que a duração do pulso de saída (T) e suficiente para permitir que a carga do capacitor C atinja o valor final.

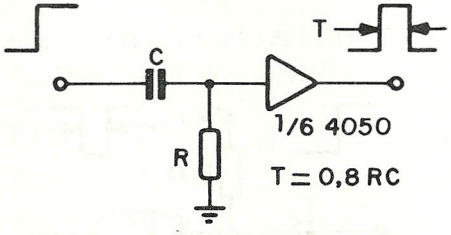
A figura 6.19 ilustra os detetores de borda com circuitos integrados CMOS. Nestes circuitos é necessário utilizar CI'S da série B para conseguir pulsos com tempos de subida e descida curtos:



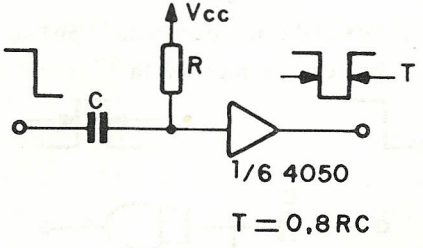
(a)Subida na entrada, pulso negativo na saída



(b)Descida na entrada, pulso positivo na saída



(c)Subida na entrada, pulso positivo na saída



(d)Descida na entrada, pulso negativo na saída

Fig. 6.19 - Detetores de Borda com CMOS.

Os circuitos mostrados nas figuras 6.18 e 6.19 são úteis quando se deseja gerar um pulso de saída como resposta a uma transição na entrada. A duração do pulso fornecido na saída do circuito não é muito precisa, pois depende das características de cada porta individual e varia com a temperatura. Contudo, quando a duração precisa não for importante e quando os requisitos para a entrada puderem ser atendidos (duração maior que o pulso de saída, tempo suficiente entre transições), os detetores de borda são um recurso simples e eficiente para geração de pulsos.

O temporizador 555 pode ser utilizado também como monoestável, conforme ilustra a figura 6.20:

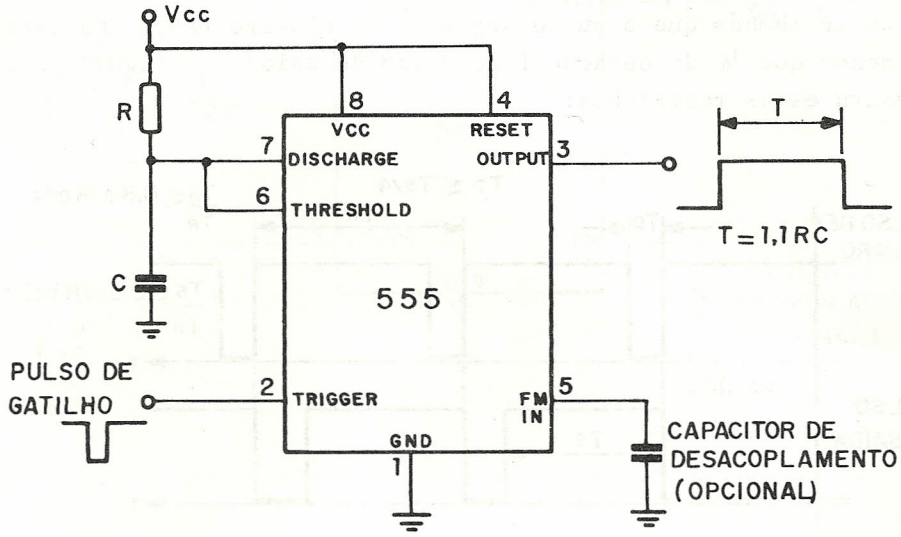


Fig. 6.20 - Monoestável com 555.

Neste circuito, a saída está normalmente em "Ø" e o transistor de descarga interno ao 555 está conduzindo, mantendo C descarregado. Aplicando um pulso de gatilho na entrada TRIGGER, conforme ilustra a figura, a saída vai para o estado lógico "1" e o transistor de descarga pára de conduzir. O capacitor C se carrega através de R até que sua tensão alcance $^2/_3$ de V_{cc}. Quando isto ocorre, o comparador (1) do 555 seta o FF de

controle e o transistor de descarga conduz, descarregando rapidamente o capacitor. Ao mesmo tempo, a saída volta para o estado "Ø" e nele permanece até a ocorrência de um novo pulso de disparo.

Neste circuito a duração do pulso de disparo negativo deve ser menor que a duração T do pulso de saída. O valor máximo de R é de 3,3 MΩ e o valor mínimo é de 1KΩ; para o capacitor C recomenda-se um valor mínimo de 5ØØpF e o valor máximo é limitado pela corrente de fuga admissível. Se a entrada de gatilho recebe pulsos em seqüência, a saída terá a forma de uma onda retangular e podemos falar em um DUTY CYCLE para este sinal. Recomenda-se que o DUTY CYCLE seja inferior a 80% para valores elevados de R e inferior a 50% para valores baixos de R; recomenda-se também que o pulso negativo de disparo tenha uma largura menor que ¼ da duração T do pulso de saída. A figura 6.21 ilustra estas restrições:

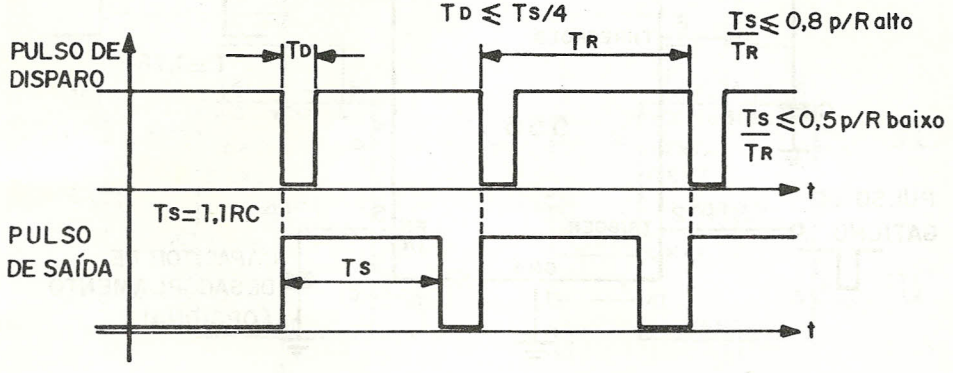


Fig. 6.21 - Formas de Onda para o Monoestável da figura 6.20 e Restrições aplicáveis.

Em geral, um monoestável requer um certo espaçamento entre os pulsos de disparo, de modo que um novo disparo só ocorra após o monoestável haver retornado a seu estado estável. Este tipo de monoestável denomina-se NON-RETTRIGGERABLE, ou seja, não redisparável.

Existem, no entanto, monoestáveis que podem ser disparados novamente antes que a saída retorne à condição original; A cada pulso de disparo começa novamente a contar-se o tempo de duração do pulso de saída. Estes monoestáveis são denominados RETTRIGGERABLE ou redisparável.

A figura 6.21 ilustra o circuito de um monoestável redisparável construído com o 555;

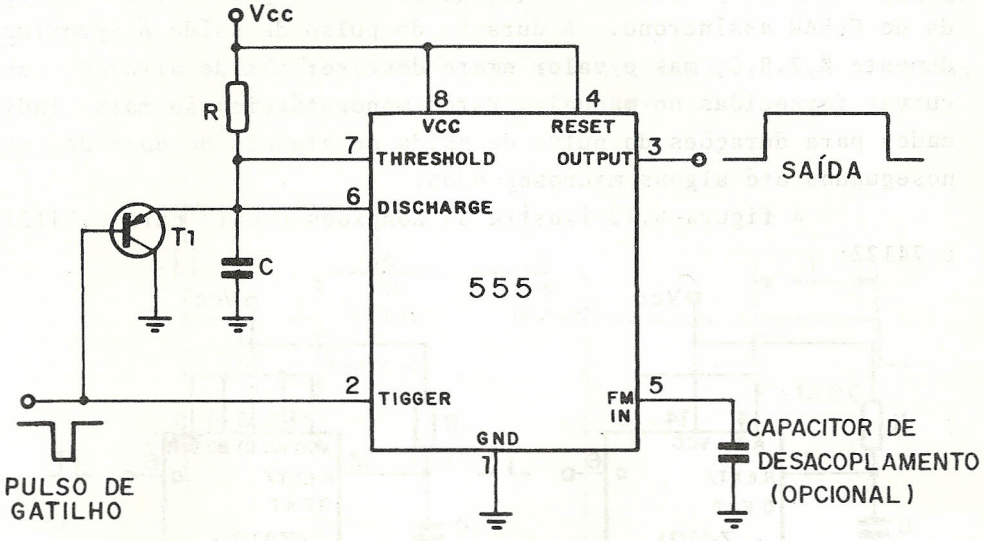


Fig. 6.21 - Monoestável Redisparável com Temporizador 555.

No circuito da figura 6.21, se ocorre um novo pulso de disparo antes de decorrido o intervalo de 1,1R.C desde a aplicação do pulso anterior, o capacitor C se descarrega rapidamente através de T_1; quando o pulso de disparo volta para o nível "1", o capacitor C começa novamente a carregar-se e a saída continua em "1" por um novo intervalo 1,1R.C, no mínimo.

Uma aplicação imediata para os monoestáveis redisparáveis é na deteção de um pulso ausente em um trem de pulsos. Vamos supor que seja aplicado à entrada de disparo do circuito da figura 6.21 um trem de pulsos negativos cujo espaçamento seja

1ØØ μs. Fazendo a duração do pulso de saída igual a, por exemplo, 15Ø μs, a saída permanecerá em "1" enquanto o monoestável estiver recebendo os pulsos de disparo a intervalos de 1ØØ μs. Se um pulso de disparo estiver ausente, a saída voltará para "Ø", indicando a falta deste pulso.

Há alguns monoestáveis disponíveis na forma de circuitos integrados TTL. Entre estes podemos citar o 74121, que não é redisparável, e o 74122, redisparável. Estes monoestáveis possuem saídas complementares, Q e $\overline{Q}$, e o 74122 possui uma entrada de CLEAR assíncrono. A duração do pulso de saída é aproximadamente Ø,7.R.C, mas o valor exato deve ser obtido através de curvas fornecidas no manual. Estes monoestáveis são mais indicados para durações do pulso de saída de algumas dezenas de nanosegundos até alguns microsegundos.

A figura 6.22 ilustra as conexões usuais para o 74121 e 74122:

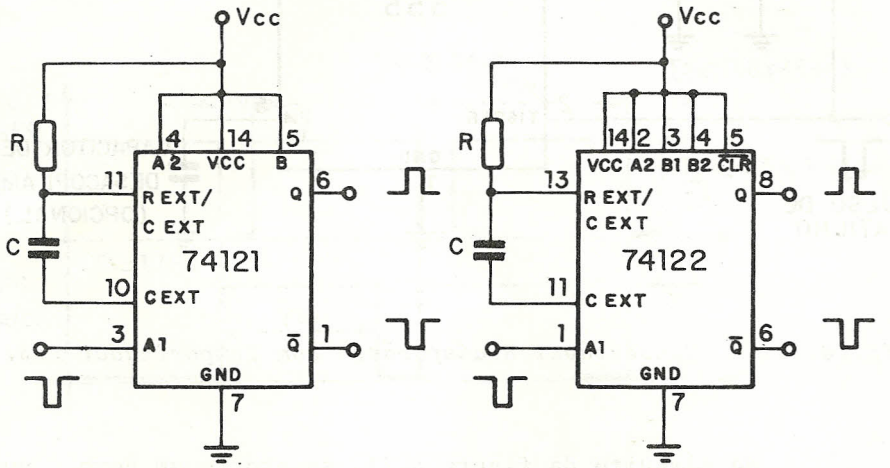


Fig. 6.22 - Conexões do 74121 e 74122.

Diversas outras ligações são possíveis, permitindo o disparo através de um pulso positivo ou através de uma rampa. O manual deve ser consultado para maiores detalhes. Há também uma versão dual do 74122, que contém dois monoestáveis independentes; é o 74123.

Na família CMOS encontramos também alguns monoestáveis.

Um exemplo é o 4Ø98, que contém dois monoestáveis independentes e redisparáveis. A figura 6.23 mostra as conexões de um monoestável 4Ø98 para disparo com pulsos positivos ou negativos:

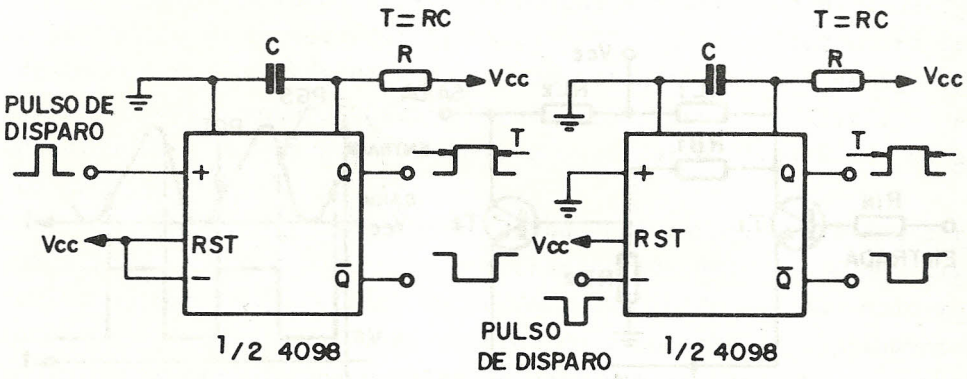


Fig. 6.23 - Conexões do 4Ø98 para disparo com pulsos positivos e negativos.

O 4Ø98 possui uma particularidade que limita sua aplicação em certos casos. O capacitor C é descarregado sempre após cada pulso de disparo e esta característica provoca um atraso entre o momento da aplicação do pulso de disparo e a transição da saída. Este atraso pode variar entre centenas de nanosegundos até vários milisegundos, para capacitores de microfarads. Para reduzir ao mínimo este atraso, é interessante utilizar o maior valor possível para R e o menor valor possível para C, para uma dada constante de tempo. O circuito integrado não pode ser utilizado quando se requer uma saída imediatamente após o disparo.

6.6 - Disparadores SCHMITT

Um circuito de grande aplicação prática é o chamado disparador SCHMITT (SCHMITT TRIGGER). O disparador SCHMITT é utilizado como restaurador de formas de onda, tendo em vista

uma característica especial que apresenta, denominada histerese. A figura 6.24 ilustra o circuito de um disparador SCHMITT com componentes discretos e mostra também formas de onda de entrada e saída que esclarecem seu funcionamento:

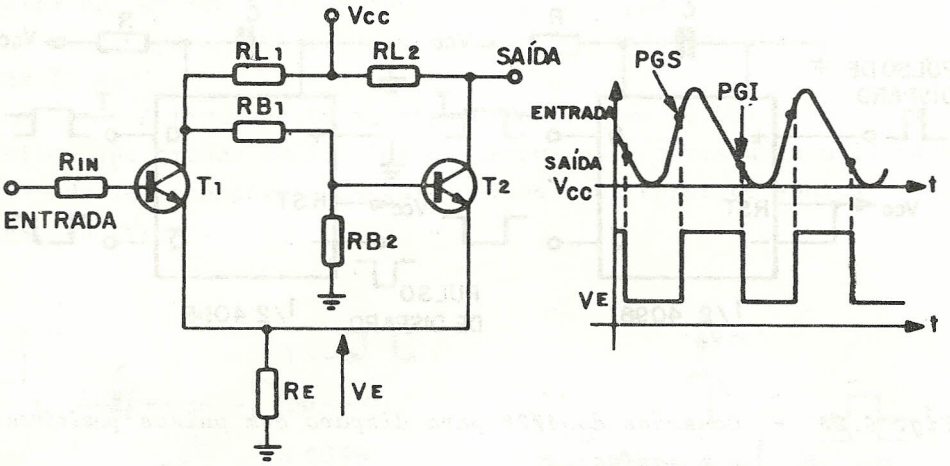
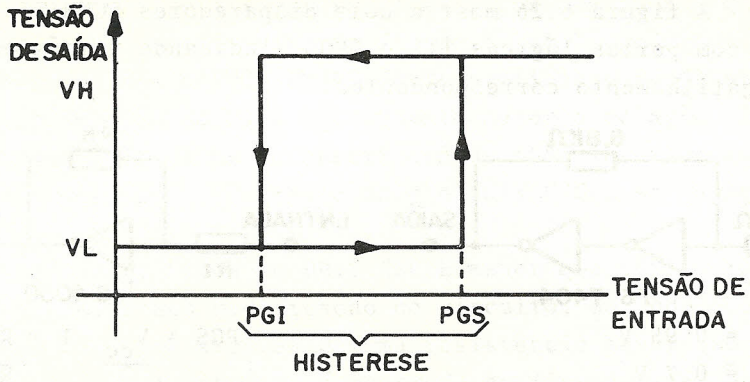


Fig. 6.24 - Disparador Schmitt com Componentes Discretos e Formas de Onda.

Vamos admitir que, inicialmente, T_1 está cortado e T_2 conduzindo. Quando a tensão na entrada atinge um valor PGS, ou ponto de gatilhamento superior (UPPER TRIGGER POINT) T_1 passa a conduzir e a realimentação do circuito faz com que T_2 vá para o corte e a tensão de saída fique próxima de V_{CC}. Diminuindo a tensão de entrada, quando ela atinge o valor PGI, ou ponto de gatilhamento inferior (LOWER TRIGGER POINT) T_1 deixa de conduzir e T_2 vai para a saturação, de modo que a tensão de saída assume um valor próximo a V_E. O circuito do disparador SCHMITT é projetado de modo que as tensões PGS e PGI sejam diferentes, a diferença PGS-PGI denomina-se histerese do circuito, por analogia ao que ocorre no fenômeno de magnetização do ferro. A figura 6.25 ilustra a curva de transferência (tensão de saída como função da tensão de entrada) para um disparador SCHMITT.



V_H = tensão de saída no estado lógico "1"

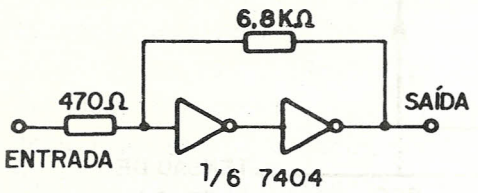
V_L = tensão de saída no estado lógico "Ø".

Fig. 6.25 - Curva de Transferência de um Disparador Schmitt.

Uma característica inerente ao disparador SCHMITT e a maior imunidade ao ruído, decorrente da histerese. Observando a figura 6.25, vemos que, uma vez que a tensão de entrada tenha atingido o PGS, a saída se manterá em V_H, mesmo que haja um ruído superposto à forma de onda, desde que a amplitude deste ruído seja menor que a histerese. De modo análogo, uma vez que a entrada tenha atingido o PGI, a saída ficará em V_L, ignorando um ruído de amplitude menor que a histerese do circuito.

Num circuito digital as entradas devem possuir tempos de subida e descida curtos. Caso contrário, o circuito pode tornar-se instável, devido aos transistores internos ao CI ficarem muito tempo na região ativa, e elementos que respondem a transições do sinal de entrada (EDGE SENSITIVE) podem deixar de operar. A realimentação positiva existente no disparador SCHMITT permite converter as variações lentas na entrada em transições rápidas na saída, conforme ilustra a conversão da senoide em uma onda quadrada na figura 6.24. Assim, o disparador Schmitt é indicado para interface entre sinais de variação lenta e circuitos lógicos.

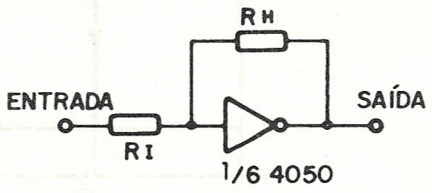
A figura 6.26 mostra dois disparadores SCHMITT construídos com portas lógicas TTL e CMOS, indicando também os pontos de gatilhamento correspondentes:



PGS = 0,95 V

PGI ≅ 0,7 V

(a) com portas TTL



$$PGS = \frac{V_{CC}}{2}\left(1 + \frac{R_I}{R_H}\right)$$

$$PGI = \frac{V_{CC}}{2}\left(1 - \frac{R_I}{R_H}\right)$$

(b) com portas CMOS

Fig. 6.26 - Disparadores SCHMITT com portas lógicas TTL e CMOS.

No circuito da figura 6.26(a) é necessário que o sinal de entrada provenha de uma fonte de baixa impedância, no circuito da figura 6.26(b) a impedância de entrada está limitada pelo valor de R_I.

Uma solução mais conveniente na maioria dos casos é utilizar os disparadores SCHMITT já disponíveis na forma de circuitos integrados.

Na série 74TTL encontramos, por exemplo, o 7413, que contém duas portas NAND de quatro entradas com características de disparador SCHMITT, ou o 7414, que contém seis inversores do tipo SCHMITT TRIGGER. Em ambos os casos PGI ≅ Ø,9V e PGI≅1,7V, com uma histerese da ordem de Ø,8V.

Na série 4ØØØ CMOS podemos citar o 4Ø93, contendo quatro portas NAND de duas entradas com características de disparador SCHMITT. Para V_{CC}=5V, o 4Ø93 tem PGI = 2,3V e PGS = 2,9V, com uma histerese de Ø,6V, para V_{CC}=1Ø V, PGI=3,9V e PGS=5,9V, com histerese de 2 V. A impedância de entrada é essencialmente infinita.

A utilização dos disparadores SCHMITT indicados acima (7413, 7414, 4Ø93) permite alcançar melhores resultados em termos de velocidade e impedância de entrada e dispensa o uso de

resistores externos, sendo por isto preferível aos circuitos da figura 6.26 na maior parte das aplicações.

Em todos os casos, as portas não utilizadas como disparadores SCHMITT podem ser empregadas normalmente.

6.7 - Distribuição de Clocks

Em itens anteriores, vimos alguns circuitos osciladores capazes de gerar o sinal de CLOCK para um sistema digital.

Quando se trata de um pequeno sistema digital e/ou o CLOCK seja de baixa frequência não há grandes problemas na distribuição do mesmo para todo o circuito.

Entretanto, num sistema com grande número de elementos que requeiram o sinal de CLOCK, este deve ser distribuído de modo a respeitar as regras de carregamento (FAN-OUT) aplicáveis. Pode ocorrer que os diversos elementos que necessitam de CLOCK estejam em placas diferentes e que o comprimento físico das linhas de distribuição sejam distintos. A figura 6.27 ilustra esta situação:

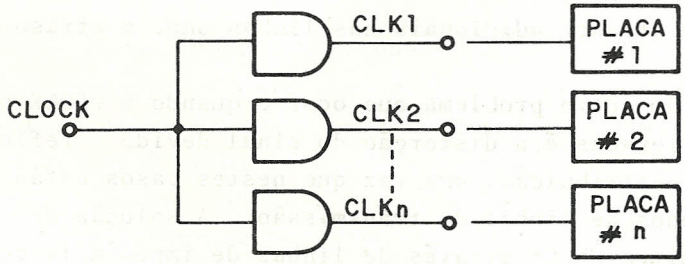


Fig. 6.27 - Distribuição do Sinal de CLOCK para diversas Placas em um sistema digital.

É obvio que os atrasos introduzidos pelos diversos caminhos serão distintos e isto acarreta um fenômeno conhecido como CLOCK SKEW, ilustrado na figura 6.28:

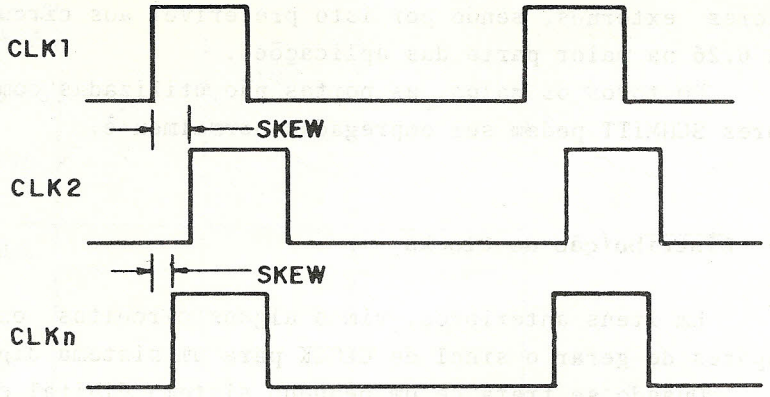


Fig. 6.28 - Defasagem entre CLOCKS.

Esta defasagem ou SKEW entre os diversos sinais de CLOCK pode acarretar problemas sérios no funcionamento do sistema, principalmente se a freqüência do CLOCK for elevada e as diferenças entre os sinais representarem um valor significativo em relação ao período ou duração do pulso. É importante que o SKEW seja reduzido ao mínimo aceitável o que é feito geralmente introduzindo atrasos adicionais nas linhas onde o atraso total é menor.

Um outro problema que ocorre quando o CLOCK possui freqüências elevadas é a distorção do sinal devido a reflexões nas linhas de distribuição, uma vez que nestes casos estão envolvidos fenômenos de linhas de transmissão. A solução do problema é distribuir o CLOCK através de linhas de impedância conhecida, geralmente cabos coaxiais, e com terminação correta. Freqüentemente utilizam-se amplificadores em coletor comum nas saídas de CLOCK, com o resistor de terminação colocado no extremo da linha de distribuição, conforme ilustra a figura 6.29:

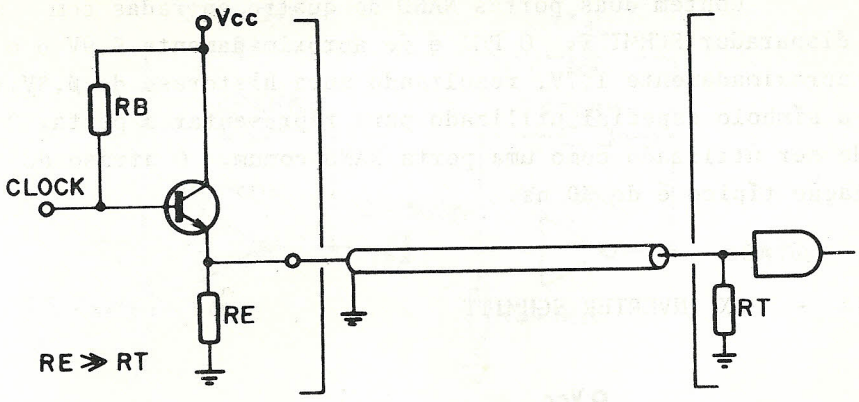


Fig. 6.29 - Distribuição de CLOCK através de linha Coaxial de Impedância R_T.

6.8 - Monoestáveis e Disparadores SCHMITT da Família TTL

Neste item apresentamos alguns monoestáveis e disparadores SCHMITT da série 74TTL com suas características principais.

7413 - DUAL 4-INPUT NAND SCHMITT

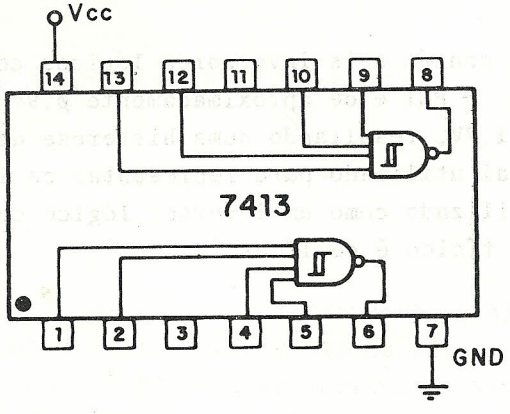


Fig. 6.30 - 7413.

Contém duas portas NAND de quatro entradas com ação de disparador SCHMITT. O PGI é de aproximadamente Ø,9V e o PGS de aproximadamente 1,7V, resultando numa histerese de Ø,8V. Observe o símbolo especial utilizado para representar a porta. O 7413 pode ser utilizado como uma porta NAND comum. O atraso de propagação típico é de 30 ns.

7414 - HEX INVERTER SCHMITT

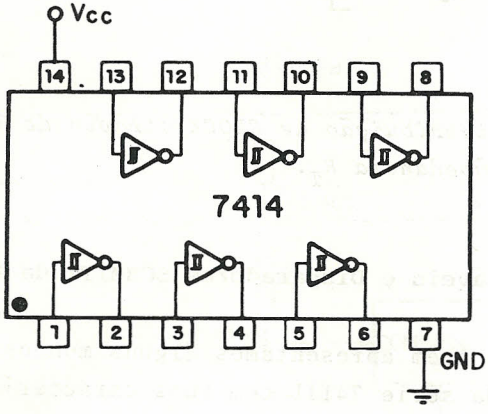


Fig. 6.31 - 7414.

O 7414 contém seis inversores lógicos com ação de disparador SCHMITT. O PGI é de aproximadamente Ø,9V e o PGS de aproximadamente 1,7V, resultando numa histerese de Ø,8V. Observe o símbolo especial utilizado para representar os inversores. O 7414 pode ser utilizado como um inversor lógico comum. O atraso de propagação típico é de 17 ns.

74121 - MONOSTABLE MULTIVIBRATOR, NOT RETTRIGGERABLE

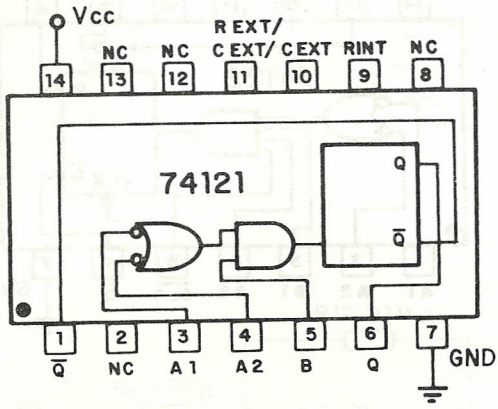


Fig. 6.32 - 74121.

Este circuito integrado contém um multivibrador monoestável não redisparável. A duração do pulso de saída é determinada por um capacitor C conectado entre os pinos 1Ø e 11 e um resistor R conectado entre o pino 11 e V_{CC}. O valor de R pode variar entre 2KΩ e 4ØKΩ e o de C deve ser maior que 1Ø pF. As entradas A_1, A_2 e B são entradas de disparo e podem ser utilizadas de diversas maneiras. Por exemplo, se A_1=A_2="Ø", uma transição positiva na entrada B dispara o monoestável. Se A_1=B="1", uma transição negativa na entrada A_2 provoca o disparo; se A_2=B="1", uma transição negativa em A_1 dispara o monoestável. A entrada B tem ação de disparador SCHMITT e pode ser utilizada para disparar o monoestável a partir de sinais com tempos de transição elevados, como uma rampa, por exemplo. As entradas de disparo não utilizadas devem ser ligadas ao nível lógico adequado e nunca deixadas abertas.

74122 - MONOESTABLE MULTIVIBRATOR, RETTRIGGERABLE

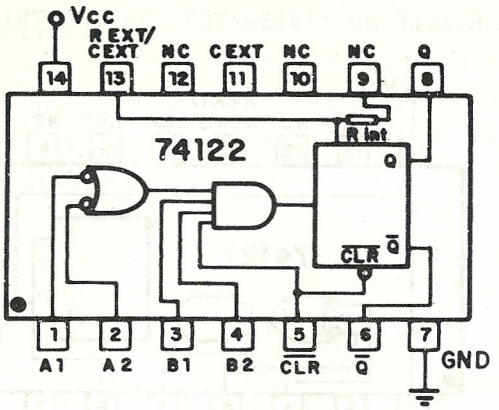


Fig. 6.33 - 74122.

O 74122 contém um monoestável redisparável com entrada de CLEAR assíncrono. A duração do pulso de saída é determinada por um capacitor C conectado entre os pinos 11 e 13 e um resistor R conectado entre o pino 13 e V_{cc}. O valor de R deve estar situado entre 5KΩ e 25KΩ e o de C deve ser maior que 1ØpF.

Para operação normal, a entrada $\overline{CLR}$ deve ficar em "1"; $\overline{CLR}$="Ø" inibe o disparo e força Q="Ø", $\overline{Q}$="1". As entradas A_1, A_2, B_1 e B_2 são para disparo do monoestável, o que pode ser conseguido de diversos modos. Por exemplo, se $A_1 = A_2$ = "Ø" e B_2= "1", uma transição positiva em B_1 dispara o circuito; se $A_1=B_1=B_2$="1", uma transição negativa em A_2 provoca o disparo. As entradas de disparo não utilizadas devem ser mantidas no nível lógico adequado e nunca deixadas abertas.

6.9 - Monoestáveis e Disparadores SCHMITT da Família CMOS

Apresentamos a seguir alguns circuitos integrados da série 4ØØØ CMOS que realizam a função de monoestáveis e disparadores SCHMITT.

4Ø93 - QUAD 2-INPUT NAND SCHMITT

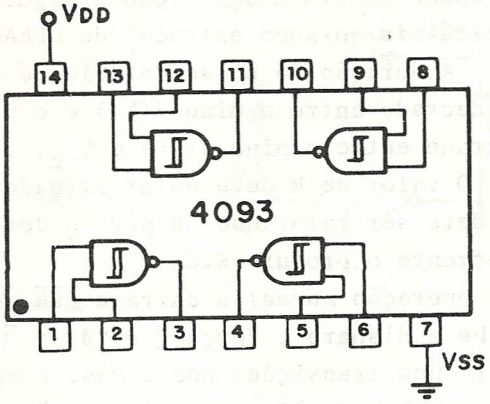


Fig. 6.34 - 4Ø93.

Este circuito integrado contém quatro portas NAND de duas entradas com ação de disparador SCHMITT. Para V_{CC}=5V, tem-se PGI=2,3V e PGS=2,9V, com uma histerese de Ø,6V; para V_{CC}=1ØV, tem-se PGI=3,9V e PGS=5,9V, com uma histerese de 2,ØV. Observe o símbolo especial usado para representar a porta. O 4Ø93 pode ser empregado como uma porta NAND comum. O atraso de propagação típico é de 3ØØ ns para V_{CC}=5V e 15Ø ns para V_{CC}=1ØV.

4Ø98 - DUAL RETTRIGGERABLE MONOSTABLE

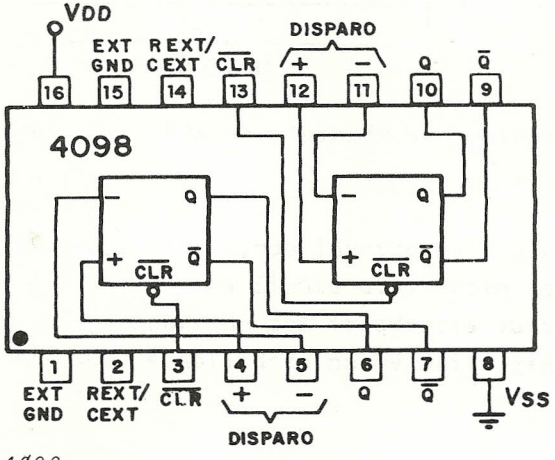


Fig. 6.35 - 4Ø98.

O 4Ø98 contém dois multivibradores monoestáveis completamente independentes. A descrição a seguir se aplica a ambos. Os monoestáveis possuem entradas de CLEAR assíncrono e são redisparáveis. A duração do pulso de saída é determinada por um capacitor C conectado entre o pino 2(14) e o pino 1(15) e um resistor R conectado entre o pino 2(14) e V_{cc}; o pino 1(15) deve ser aterrado. O valor de R deve estar situado entre 1ØKΩ e 1ØMΩ e o valor de C deve ser maior que 2Ø pF. A duração do pulso de saída é simplesmente o produto R.C.

Para operação normal a entrada $\overline{CLR}$ deve ficar em "1", $\overline{CLR}$ em "Ø" inibe o disparo e força Q = "Ø" e $\overline{Q}$ = "1". Para disparar o circuito nas transições positivas, a entrada (-) deve ser aterrada e o pulso de disparo aplicado à entrada (+); caso se deseje o gatilhamento nas transições negativas, a entrada (+) deve ser conectada a V_{cc} e o pulso de disparo aplicado à entrada (-).

A largura mínima do pulso de disparo é de 7Ø ns com V_{cc}=5V e 3Ø ns para V_{cc}=1ØV; a largura mínima do pulso de saída é de 55Ø ns com V_{cc}=5V e 35Ø ns com V_{cc}=1ØV.

BIBLIOGRAFIA

Livros:

IDOETA, Ivan Valeije e CAPUANO, Francisco Gabriel - *Elementos de Eletrônica Digital*, 6ª Edição, São Paulo, Livros Érica Editora Ltda, 1983.

LANCASTER, Donald - *CMOS COOKBOOK*, 1ª Edição, 6ª Impressão, Indianapolis, Howard W. Sams & Cv.,Inc., 1981.

LANCASTER, Donald - *TTL COOKBOOK*, 1ª Edição, 12ª Impressão, Indianapolis, Howard W. Sams & Cv.,Inc., 1982.

LANDO, Roberto Antonio e ALVES, Serg Rios - *Amplificador Operacional*, 1ª Edição, São Paulo, Livros Érica Editora Ltda, 1983.

TEXAS INSTRUMENTS INC. - *Designing With TTL Integrated Circuits*, New York, McGraw-Hill Book Company, 1971.

Manuais:

CMOS Integrated Circuits - Harris Semiconductor.

System 74 Designer's Manual - Texas Instruments Inc.

The TTL Data Book - Texas Instruments Inc.

PUBLICAÇÕES ÉRICA

LINGUAGENS E APLICATIVOS

Alexandre	- AutoCad 10.0 - Guia Prático
Arsonval	- Basic para Computadores Pessoais
Ideali	- Sistema Operacional CP/M - 80
Ivan	- Wordstar auto explicativo
Mesquita	- Linguagem C
Natale	- Basic Avançado
Natale	- Probasic - Programação em Basic
Santos	- Aplicativos
Sanches	- OrCAD - Ref. e Consulta
Mesquita	- Turbo Pascal - Teoria e Programas
Mota	- como programar em d'Base III
Mota	- d'Base III - Plus - Guia Prático
Sgambati/Penna	- PC Sistema Mestre
Mota	- FoxBase - Guia Prático
Lanto	- Como Informatizar seu Empreendimento
W. Leite	- AutoLisp
P.R.P.	- Ventura Publisher
Microcenter	- Procedures ferramentas de programação FoxBase +; dBase III Plus, Clipper Summer 87

MICROPROCESSADORES

Ideali	- Microprocessador 68000 - Hardware - Vol. 1
Ideali	- 68000 Família de Microprocessadores - 32 bits - Software - Vol. 2
Cypriano/Cardinali	- Microprocessadores Z-80 - Hardware - Vol. 1
Cypriano	- Microprocessadores Z-80 - Software - Vol. 2
Vidal	- Microcontroladores
Visconti	- Microprocessadores 8080 e 8085 - Hardware - Vol. 1
Visconti	- Microprocessadores 8080 e 8085 - Software - Vol. 2
Carvalho/Rocha	- 8086/8088 - Software - Vol. 1

ELETRÔNICA

Albuquerque	- Análise de Circuitos em Corrente Contínua
Albuquerque	- Análise de Circuitos em Cor. Alternada
Almeida	- Eletrônica Industrial
Almeida	- Eletrônica de Potência
Azevedo	- TTL/CMOS - Circuitos Integrados - Vol. 1
Azevedo	- TTL/CMOS - Circuitos Integrados - Vol. 2
Capuano/Idoeta	- Elementos de Eletrônica Digital
Capuano/Marino	- Laboratório de Eletricidade e Eletrônica
Cipelli/Sandrini	- Teoria e Desenvolvimento de Projetos de Circuitos Eletrônicos
Cuocolo	- Periféricos
Lando/Serg	- Amplificador Operacional
Mello	- Projetos de Fontes Chaveadas
Bento	- Sistemas de Controle - Teoria e Projetos
Shibata	- Elet. Digital - Teoria e Experiência - Vol. I

PROCESSAMENTO DE DADOS

Aghazarm/Miranda	- Transmissão de Dados em Sistemas de Computação
Penna	- Teleprocessamento BSC-3

TELECOMUNICAÇÕES

Gomes	- Telecomunicações - Transmissão e Recepção AM - FM - Sistemas Pulsados
Smit	- Linhas de Comunicação
Smit	- Microondas
Smit	- Ondas e Antenas
Smit	- Rádio Astronomia
Smit	- Rádio Propagação
Megrich	- TV P/B - Transmissão e Recepção - Vol. I
Antunes	- Videocassete - Teoria e Prática

MECÂNICA

Melconian	- Mecânica Técnica e Resistência dos Materiais
Filkauskas/Guedes	- O Plástico
Canale	- Automobilística

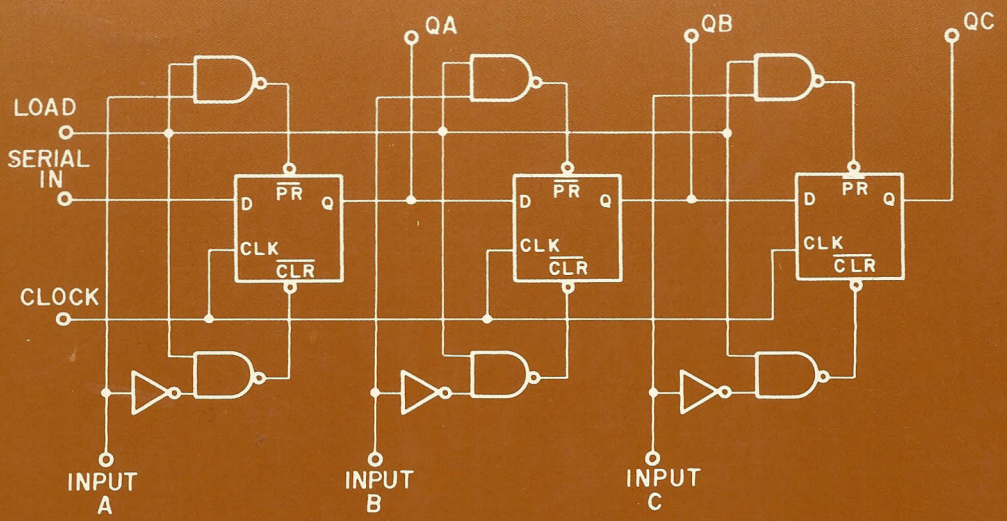
ISBN - 85-7194-015-0

JOÃO BATISTA DE AZEVEDO JÚNIOR

TTL/CMOS

TEORIA E APLICAÇÃO EM CIRCUITOS DIGITAIS

4ª EDIÇÃO



VOLUME 2

ÉRICA

CAPÍTULO 7 - CONTADORES DIGITAIS

7.1 - Introdução

Um contador binário é um circuito capaz de contar, segundo uma determinada seqüência, o número de pulsos que recebe em sua entrada. Os contadores encontram enorme aplicação em circuitos digitais executando funções como divisão de freqüência, seqüenciamento de operações, manipulações matemáticas,etc.

Podemos classificar os contadores digitais quanto ao sistema de aplicação do CLOCK em:

- contadores assíncronos: são aqueles nos quais o CLOCK é aplicado ao primeiro estágio; os estágios seguintes utilizam como CLOCK a saída do estágio anterior. Estes contadores são conhecidos também como RIPPLE COUNTERS;

- contadores síncronos: são aqueles nos quais o sinal de CLOCK é aplicado simultaneamente a todos os estágios.

Os contadores assíncronos são em geral mais simples que os síncronos, mas estes últimos possibilitam o uso de maiores freqüências de CLOCK, conforme veremos no decorrer deste capítulo.

Quanto ao modo de contagem, podemos classificar os contadores em:

- progressivos (UP COUNTERS): são aqueles que contam numa seqüência crescente;
- regressivos (DOWN COUNTERS): são aqueles que contam em uma seqüência decrescente.

Os contadores podem ser construídos utilizando FLIP-FLOPS e portas lógicas, mas, em muitos casos, é mais interessante aproveitar aqueles disponíveis na forma de circuitos integrados TTL e CMOS.

Neste capítulo, estudaremos as duas alternativas, detalhando as características, projeto e aplicações dos contadores digitais.

7.2 - Contadores Assíncronos

A figura 7.1 mostra o esquema de um contador binário de 3 bits e as formas de onda correspondentes. Nesta figura admitimos, que os três FF's estão inicialmente ressetados e que o gatilhamento ocorre na transição negativa do pulso de CLOCK:

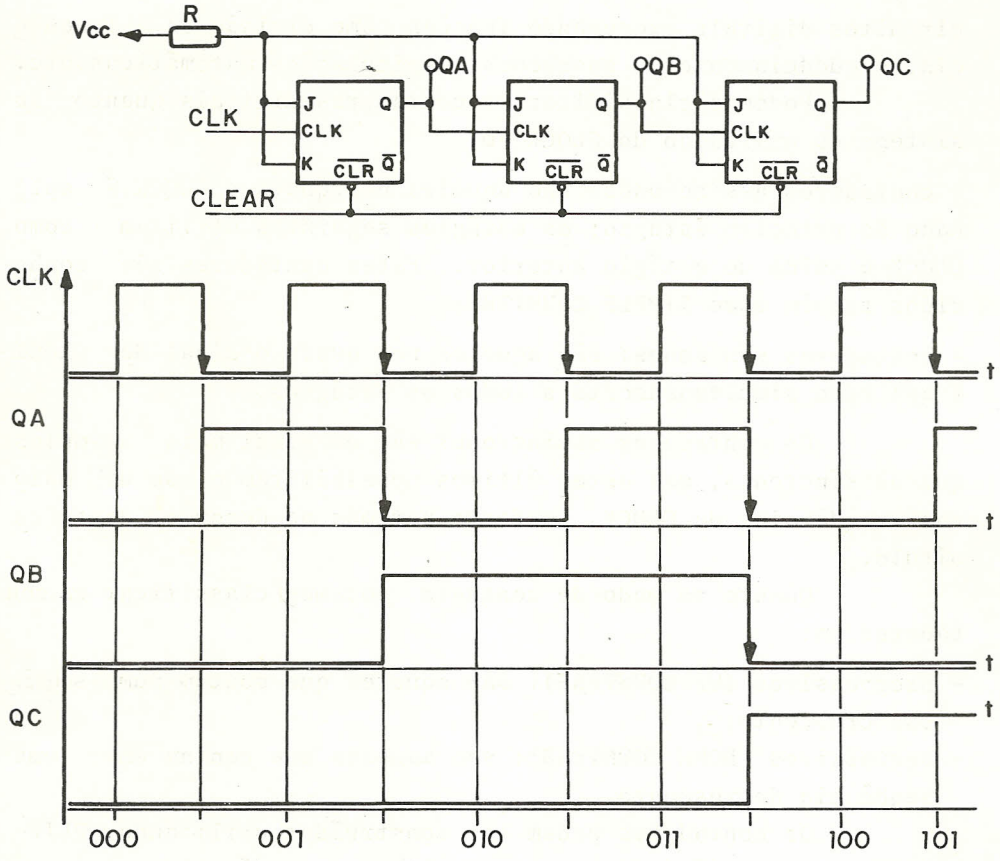


Fig. 7.1 - Contador Assíncrono de 3 Bits e Formas de Onda Correspondentes.

Vejamos como funciona o circuito. Inicialmente, Q_A=

Q_B=Q_C="Ø". Na transição negativa do primeiro pulso de CLOCK, Q_A vai para "1"; o segundo FLIP-FLOP não é afetado, pois somente as transições negativas do CLOCK o fazem mudar de estado e, neste caso, ocorreu uma transição positiva. Na transição negativa do segundo pulso de CLOCK, Q_A volta para "Ø" e Q_B vai para "1", pois agora houve uma transição negativa em sua entrada de CLOCK. Segundo este raciocínio, é fácil concluir que, de fato, o circuito da figura 7.1 irá contar os pulsos de CLOCK recebidos, segundo a seqüência ØØØ, ØØ1, Ø1Ø, Ø11, 1ØØ, 1Ø1, 11Ø e 111. Quando Q_A=Q_B=Q_C="1", a próxima transição negativa do pulso de CLOCK fará o contador retornar ao estado inicial, com Q_A=Q_B=Q_C = "Ø". Cada um dos passos da contagem é chamado de estado e podemos dizer que o circuito da figura 7.1 possui oito estados.

Observe que a freqüência do sinal de CLOCK aparece dividida por dois na saída do primeiro estágio (Q_A), por quatro na saída do segundo estágio (Q_B) e por oito na saída do terceiro estágio (Q_C). Veja também que, enquanto as saídas Q_A, Q_B e Q_C contam progressivamente, as saídas $\overline{Q}_A$, $\overline{Q}_B$ e $\overline{Q}_C$ contam regressivamente, conforme ilustra a figura 7.2:

UP	Q_C	Q_B	Q_A	$\overline{Q}_C$	$\overline{Q}_B$	$\overline{Q}_A$	DOWN
Ø	Ø	Ø	Ø	1	1	1	7
1	Ø	Ø	1	1	1	Ø	6
2	Ø	1	Ø	1	Ø	1	5
3	Ø	1	1	1	Ø	Ø	4
4	1	Ø	Ø	Ø	1	1	3
5	1	Ø	1	Ø	1	Ø	2
6	1	1	Ø	Ø	Ø	1	1
7	1	1	1	Ø	Ø	Ø	Ø

Fig. 7.2 - Saídas do Contador da figura 7.1.

Se acrescentarmos mais um estágio ao circuito da figura 7.1, teremos um contador binário de quatro bits, que contará na seqüência ØØØØ, ØØØ1,.....,111Ø, 1111, ou seja, terá 16 estados. De um modo geral, um contador binário puro, como o da

figura 7.1, com N estágios apresentará um ciclo de contagem com 2^N estados e a freqüência do CLOCK aparecerá dividida por 2^i na saída do i-ésimo estágio.

À medida que o número de estágios aumenta, a freqüência máxima permissível do sinal de CLOCK na entrada do contador assíncrono diminui. Isto é conseqüência do fato de cada FLIP-FLOP requerer um determinado tempo para mudar de estado, conforme vimos no capítulo 5. Se um novo pulso de CLOCK for aplicado, antes que a mudança ocasionada pelo anterior se tenha completado, o gatilhamento será errático. No pior caso, todos os FLIP-FLOPS mudam de estado e, como o gatilhamento de um estágio depende da mudança de estado do FLIP-FLOP anterior, é necessário permitir um tempo suficiente para que as mudanças se propaguem por todo o contador. Assim sendo, a freqüência máxima permissível para o sinal de CLOCK de um contador assíncrono semelhante ao da figura 7.1 será:

$$f \leq \frac{1}{N.T_p}$$

onde:

N - número de estágios

T_p - atraso de propagação de um FF

Por exemplo, se cada FLIP-FLOP do circuito da figura 7.1 possui um atraso de 50 ns, a freqüência máxima do sinal de CLOCK será:

$$f = \frac{1}{3 \times 50 \times 10^{-9}} = 6{,}67 \text{ MHZ}$$

Utilizando um contador binário puro, podemos conseguir ciclos de contagem que são potências de 2, ou seja, com 2^N estados, onde N é o número de estágios do contador. Entretanto, muitas vezes, desejamos um contador que tenha um ciclo de contagem ou módulo qualquer, por exemplo, 5, 13 ou, de maneira genérica, n. Para consegui-lo há duas alternativas básicas:

a) Construímos o contador de maneira que ao ser atingida a contagem n todos os estágios são ressetados assíncronamente, através das entradas de CLEAR;
b) Construímos o contador de maneira que, ao ser atingida a contagem n-1 e ocorrer a transição do CLOCK que não é responsável pelo gatilhamento, sejam ativadas as entradas de PRESET assíncrono; a transição seguinte do CLOCK ressetará o contador para "Ø".

Em ambos os casos, o objetivo é truncar a seqüência natural de contagem, fazendo com que o contador retorne ao estado "Ø" após o número desejado de estados. É claro que o número de estágios N necessários deve ser tal que $n \leq 2^N$.

Vamos ilustrar as duas alternativas através de exemplos.

Exemplo: 7.1

Construir um contador binário que utilize FLIP-FLOPS 7476 e que conte de "Ø" a 5, ou seja, possua 6 estados. (n = 6). Desejamos adotar a alternativa *a)* e sabemos que a entrada de CLEAR assíncrono do FLIP-FLOP é ativa no estado baixo.

Solução:

Como nosso contador deve possuir 6 estados, devemos utilizar 3 FLIP-FLOPS, pois $6 < 2^3$. A idéia da alternativa *a)* é fazer com que a contagem ocorra conforme ilustrada na figura 7.3:

contagem	Q_C	Q_B	Q_A
→ Ø	0	0	0
1	0	0	1
2	0	0	0
3	0	0	1
4	1	0	0
5	1	0	1
6	1	1	0

→ Este estado é instável e a linha de CLEAR assíncrona é ativada, retornando ao estado inicial.

Fig. 7.3 - Seqüência de contagem desejada para o contador do exemplo 7.1.

Uma solução para conseguir a ativação da linha de CLEAR assíncrono no momento desejado é ligar as saídas que estão em "1" na contagem 6 (Q_C e Q_B) a uma porta NAND e utilizar a saída desta porta para ressetar os FLIP-FLOPS. A figura 7.4 mostra o circuito resultante:

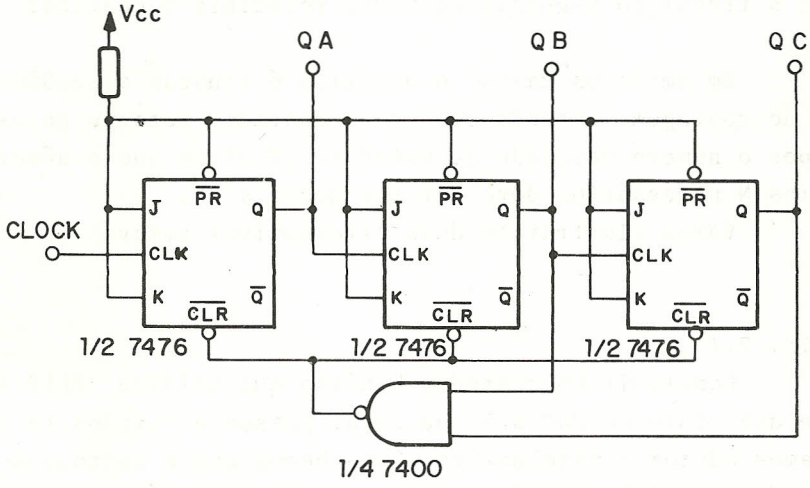


Fig. 7.4 - Contador Assíncrono divisor por 6 (módulo n=6).

As formas de onda correspondentes estão ilustradas na figura 7.5, esclarecendo o funcionamento do circuito:

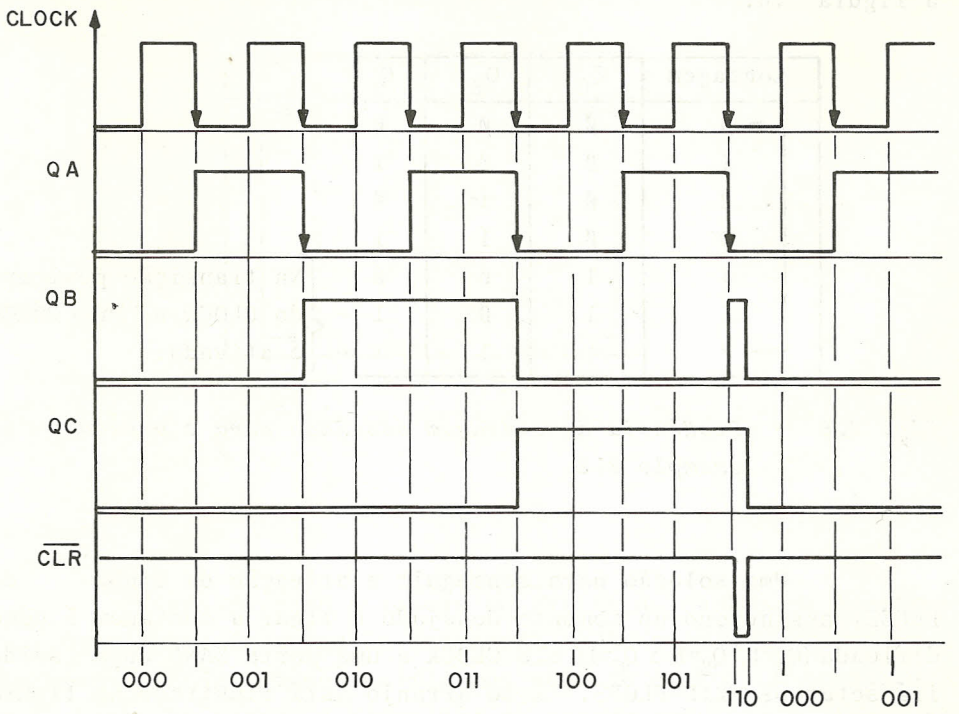


Fig. 7.5 - Formas de Onda para o Contador da figura 7.4.

Observe que nas saídas Q_B e Q_C a freqüência do CLOCK aparece dividida por 6; a saída Q_C apresenta uma forma de onda mais perfeita que a saída Q_B, pois não possui o estreito pulso decorrente do CLEAR assíncrono.

Exemplo 7.2:

Construir o contador assíncrono do exemplo anterior utilizando a alternativa b).

Solução:

Neste caso, a contagem deve ocorrer conforme ilustra a figura 7.6:

Contagem	Q_C	Q_B	Q_A	
→ Ø	Ø	Ø	Ø	
1	Ø	Ø	1	
2	Ø	1	Ø	
3	Ø	1	1	
4	1	Ø	Ø	Na transição positiva
5	1	Ø	1	← do CLOCK a linha PRESET
7	1	1	1	← é ativada.

Fig. 7.6 - Seqüência de contagem desejada para o contador do exemplo 7.2.

Uma solução para conseguir a ativação da linha de PRESET assíncrono no momento desejado é ligar a contagem 5 decodificada (Q_A=1, $\overline{Q}_B$=1 e Q_C=1) e o CLOCK a uma porta NAND cuja saída irá setar os FLIP-FLOPS. Este arranjo está ilustrado na figura 7.7:

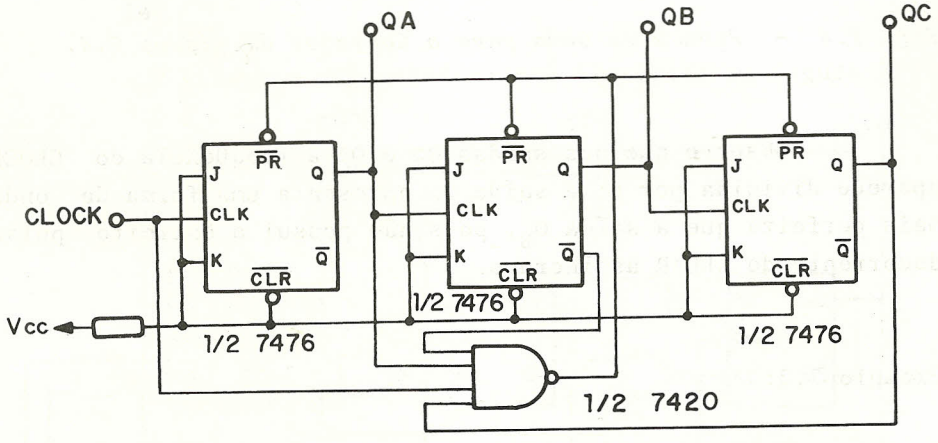


Fig. 7.7 - Contador Assíncrono divisor por 6.

As formas de onda correspondentes estão ilustradas na figura 7.8, esclarecendo o funcionamento do circuito:

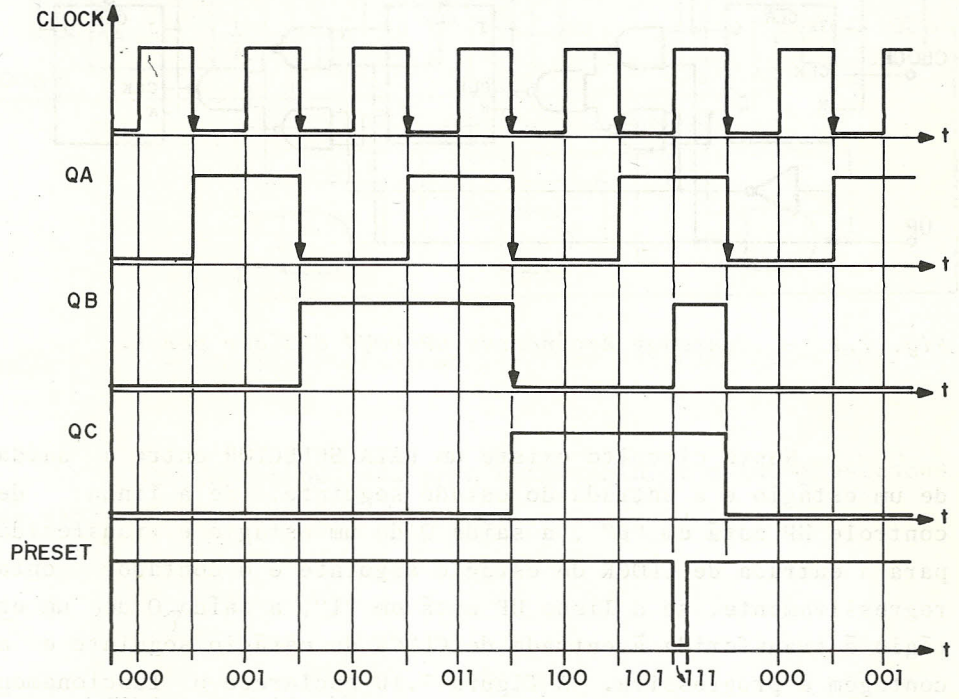


Fig. 7.8 - Formas de Onda para o Contador da figura 7.7.

Observe que na saída Q_C a freqüência do CLOCK aparece dividida por 6.

Adicionando algumas portas e uma linha de controle ao contador assíncrono, podemos construir um contador progressivo/regressivo (UP-DOWN COUNTER), ou seja, um circuito que conta progressiva ou regressivamente, conforme o estado da linha de controle. A figura 7.9 ilustra o esquema de um contador UP-

DOWN de módulo 8:

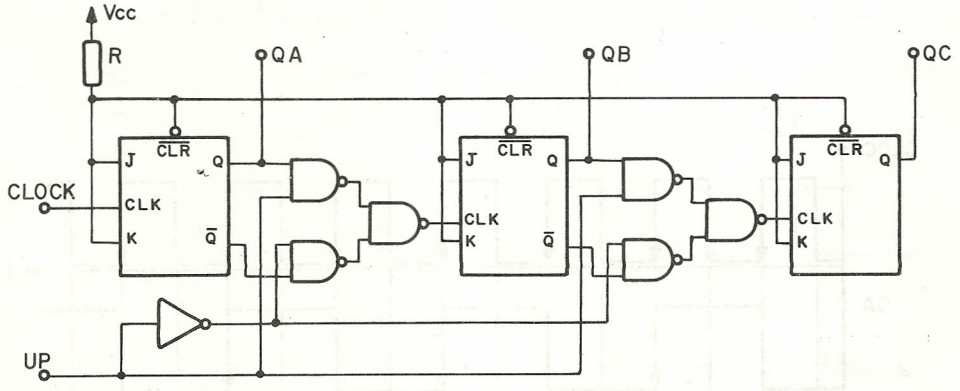


Fig. 7.9 - Contador Assíncrono UP-DOWN divisor por 8.

Neste circuito existe um DATA SELECTOR entre a saída de um estágio e a entrada do estado seguinte. Se a linha de controle UP está em "Ø" , a saída $\overline{Q}$ de um estágio é transferida para a entrada de CLOCK do estágio seguinte e o contador conta regressivamente; se a linha UP está em "1", a saída Q de um estágio é transferida à entrada de CLOCK do estágio seguinte e a contagem é progressiva. A figura 7.10 esclarece o funcionamento deste circuito, ilustrando as formas de onda correspondentes.

Para o cálculo da máxima freqüência de operação do circuito acima, devemos levar em consideração, além do atraso dos FLIP-FLOPS, os atrasos introduzidos pelas portas. Assim, para um contador, de N estágios, o período do CLOCK T deverá obedecer a seguinte equação :

$$T \geq N.T_p + (N-1).T_s$$

onde:

T_p - atraso de um FLIP-FLOP

T_s - atraso da lógica de seleção

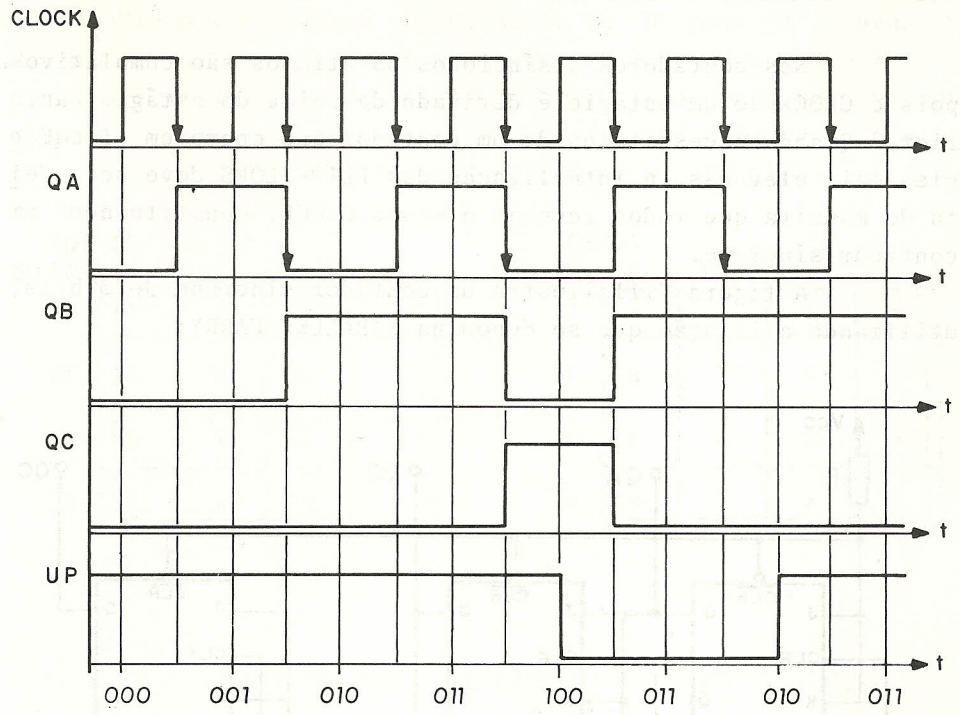


Fig. 7.10 - Formas de Onda para o contador UP-DOWN da figura 7.9.

No circuito da figura 7.9, o atraso da lógica de seleção corresponde ao de duas portas NAND. Por exemplo, se T_p=50ns e T_s=20ns (10ns para cada NAND), a freqüência máxima para o CLOCK seria

$$fmax = \frac{1}{T} = \frac{1}{(3x50+2x20)x\ 10^{-9}} = \frac{1}{190x10^{-9}}$$

$$\simeq 5,26 \text{ MHZ}$$

7.3 - Contadores Síncronos

Nos contadores assíncronos os atrasos são cumulativos, pois o CLOCK de um estágio é derivado da saída do estágio anterior. Quando necessitamos de um contador que opere em freqüências mais elevadas, a interligação dos FLIP-FLOPS deve ser feita de maneira que todos recebam o mesmo CLOCK, constituindo um contador síncrono.

A figura 7.11 ilustra um contador síncrono de 3 bits, utilizando a ligação que se denomina PARALLEL CARRY:

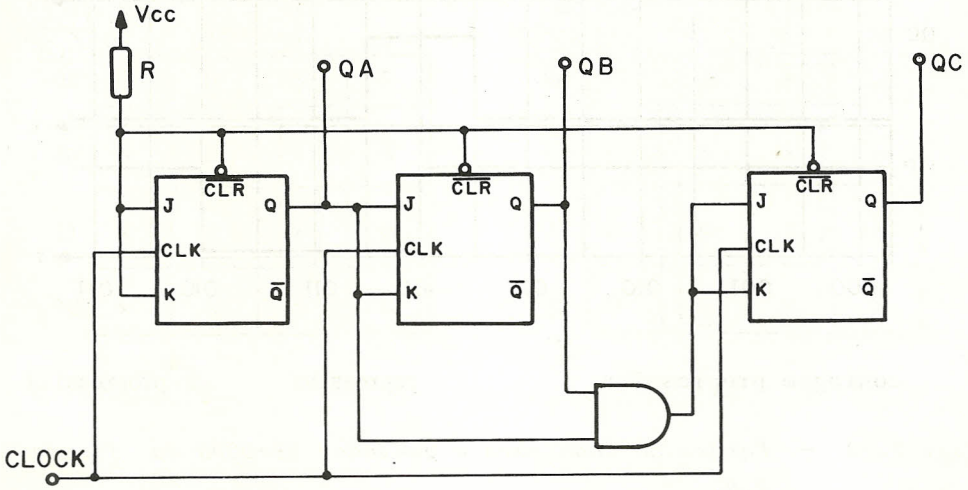


Fig. 7.11 - Contador Síncrono de 3 bits com PARALLEL CARRY.

Obserque que, agora, o controle dos FLIP-FLOPS é feito através das entradas J e K. Sabemos que se J = K = "1" o FLIP-FLOP se complementa e que se J = K = "Ø" o FLIP-FLOP não muda de estado. Assim, o primeiro estágio irá mudar de estado a cada pulso de CLOCK; O segundo só mudará de estado quando o primeiro FLIP-FLOP estiver setado, ou seja, a cada dois pulsos de CLOCK. Finalmente, o terceiro estágio só mudará de estado quando os dois primeiros estiverem em "1", ou seja, a cada quatro pulsos de CLOCK.

No circuito da figura 7.11, a freqüência máxima do CLOCK será limitada apenas pelo atraso de um FLIP-FLOP e de uma porta lógica. Isto é verdadeiro para o contador síncrono com qualquer número de estágios, utilizando a configuração de PARALLEL CARRY. O período mínimo T do sinal de CLOCK será:

$$T \geq T_p + T_g$$

onde: T_p - atraso de um FLIP-FLOP
T_g - atraso da porta lógica

Admitindo T_p = 50ns e T_g = 10ns, a freqüência máxima do sinal de CLOCK para o circuito da figura 7.11 seria:

$$f_{max} = \frac{1}{T} = \frac{1}{(50+10)\times 10^{-9}} = \frac{1}{60 \times 10^{-9}}$$

$$\cong 16{,}67 \text{ MHZ}$$

Este valor é duas vezes e meia maior que aquele calculado para o contador assíncrono da figura 7.1!

A figura 7.12 ilustra um contador síncrono de 4 bits utilizando o PARALLEL CARRY:

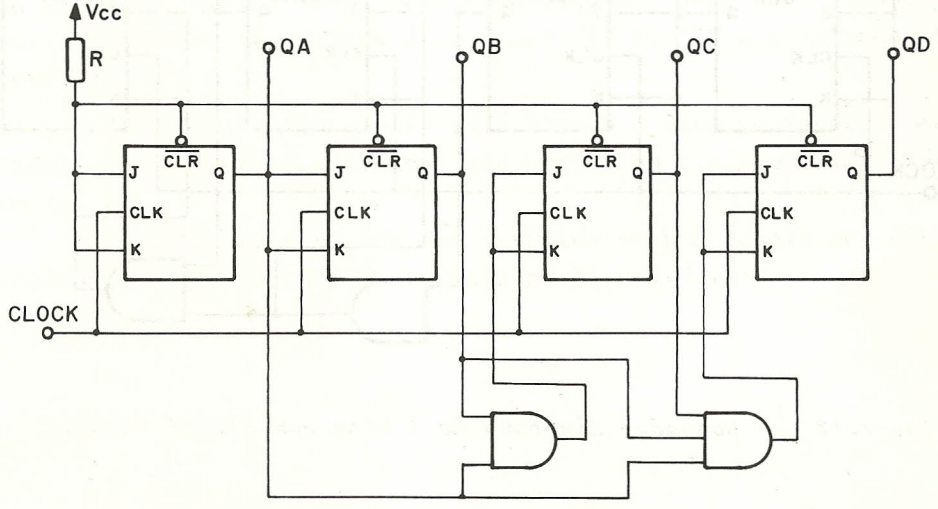


Fig. 7.12 - Contador Síncrono de 4 bits utilizando PARALLEL CARRY.

Esta figura nos permite observar um problema que ocorre nesta configuração. À medida que o número de estágios aumenta, a saída de um FF deve alimentar um número maior de portas e o número de entradas das portas torna-se maior. Este problema pode ser contornado utilizando a configuração conhecida como RIPPLE CARRY, ilustrada na figura 7.13. Nesta configuração, usamos sempre portas de duas entradas e a saída de um FF alimenta apenas uma porta. No entanto, a freqüência máxima do sinal de CLOCK será menor, pois os atrasos das portas são cumulativos. Para um contador síncrono de N estágios na configuração RIPPLE CARRY, o período mínimo do CLOCK será:

$$T \geq T_p + (N-2)\, T_g$$

onde: T_p = atraso de um FLIP-FLOP

T_g = atraso de uma porta

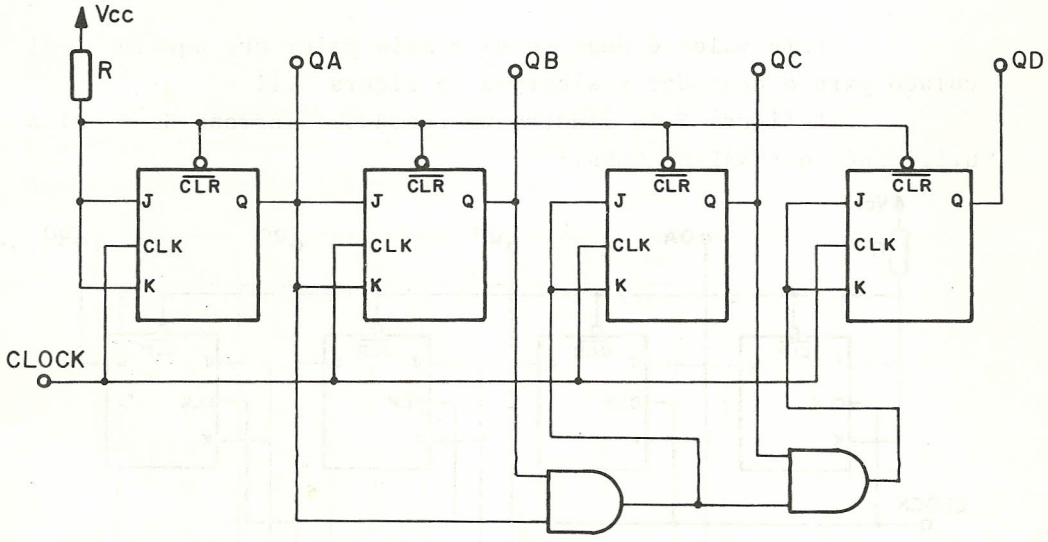


Fig. 7.13 - Contador Síncrono de 4 bits com RIPPLE CARRY.

Para T_p = 50ns e T_g = 10ns, teríamos para o circuito

da figura 7.13 uma freqüência máxima de CLOCK de

$$fmax = \frac{1}{(50+2x10) \times 10^{-9}} = \frac{1}{70x10^{-9}} \cong 14,28 \text{ MHZ}$$

7.4 - Contadores Síncronos de Seqüência Qualquer

No item anterior apresentamos os contadores síncronos de módulo igual a uma potência de dois. Algumas vezes, torna-se necessário projetar um contador de módulo diferente de 2^n ou mesmo um contador para uma seqüência qualquer.

Para fazê-lo devemos, em primeiro lugar, examinar o comportamento de um FLIP-FLOP J-K em função de seu estado presente e do estado desejado a seguir. São quatro as possibilidades:

1) Qn = "Ø" , Qn+1 = "Ø" (o FF está ressetado e deve permanecer ressetado) : neste caso, a entrada J deve ser "Ø" e a entrada K pode ser "1" ou "Ø" ;

2) Qn = "Ø", Qn+1 = "1" (o FF está ressetado e deve ser setado): neste caso, a entrada J deve ser "1" e a entrada K pode ser "1" ou "Ø" ;

3) Qn = "1", Qn+1 = "Ø" (o FF está setado e deve ser ressetado): neste caso, a entrada J pode ser "1" ou "Ø" e a entrada K deve ser "1";

4) Qn ="1", Qn+1 = "1" (o FF está setado e deve permanecer setado): neste caso, a entrada J pode ser 1 ou Ø e a entrada K deve ser Ø .

Esta análise pode ser resumida em uma tabela de excitação para o FF J-K, que é mostrada na figura 7.14:

Qn	Qn+1	J	K
Ø	Ø	Ø	X
Ø	1	1	X
1	Ø	X	1
1	1	X	Ø

Fig. 7.14 - Tabela de Excitação para o FLIP-FLOP J-K.

Uma vez estabelecida a seqüência de contagem desejada, colocamos em um mapa de Karnaugh as entradas J e K necessárias para que de um estado se passe ao estado seguinte, tendo em vista a tabela 7.14. Em seguida, retiramos as equações lógicas das entradas "J" e "K". É necessário um mapa para cada entrada de cada FF; as variáveis são as saídas do contador. Este processo é melhor esclarecido através de exemplos.

Exemplo 7.3: - Construir um contador síncrono que conte na seqüência abaixo indicada (fig. 7.15).

Solução:

O contador deve contar na seqüência conhecida como BCD 8421; trata-se de uma década contadora e devemos fazer dois mapas para cada um dos FLIP-FLOPS.

Q_D	Q_C	Q_B	Q_A	Contagem
0	0	0	0	Ø
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

Fig. 7.15 - Seqüência de contagem para o Contador do exemplo 7.3.

Consideremos o FLIP-FLOP A. Na contagem "Ø" devemos ter J_A="1" e K_A="X" para que Q_A passe de "Ø" para "1", avançando para a contagem "1". Na contagem "1" devemos ter J_A="X" e K_A="1" para que Q_A passe de "1" para "Ø", avançando para a contagem 2. Prosseguindo com este raciocínio através de toda a seqüência, construímos as tabelas para J_A e K_A; mostradas na figura 7.16:

QD QC \ QB QA	00	01	11	10
00	1	x	x	1
01	1	x	x	1
11	x	x	x	x
10	1	x	x	x

$J_A = 1$

QD QC \ QB QA	00	01	11	10
00	x	1	1	x
01	x	1	1	x
11	x	x	x	x
10	x	1	x	x

$K_A = 1$

Fig. 7.16 - Mapas de Karnaugh para as entradas J_A e K_A.

Pelos mapas da figura 7.16, vemos que o FLIP-FLOP A se comportará como desejado se as entradas J_A e K_A forem mantidas sempre em "1".

A mesma análise deve ser feita para os demais FLIP-FLOPS; os mapas e equações resultantes estão ilustrados na figura 7.17:

QD QC \ QB QA	00	01	11	10
00	0	1	x	x
01	0	1	x	x
11	x	x	x	x
10	0	0	x	x

$J_B = \overline{Q}_D \cdot Q_A$

QD QC \ QB QA	00	01	11	10
00	x	x	1	0
01	x	x	1	0
11	x	x	x	x
10	x	x	x	x

$K_B = Q_A$

QD QC \ QB QA	00	01	11	10
00	0	0	1	0
01	x	x	x	x
11	x	x	x	x
10	0	0	x	x

$J_C = Q_B \cdot Q_A$

QD QC \ QB QA	00	01	11	10
00	x	x	x	x
01	0	0	1	0
11	x	x	x	x
10	x	x	x	x

$K_C = Q_B \cdot Q_A$

QD QC \ QB QA	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	x	x	x	x
10	x	x	x	x

$J_D = Q_C \cdot Q_B \cdot Q_A$

QD QC \ QB QA	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	x	x	x	x
10	0	1	x	x

$K_D = Q_A$

Fig. 7.17 - Mapas de Karnaugh para as demais entradas J e K.

Resumindo os resultados, temos as seguintes equações:

$J_A = 1$ $\qquad$ $K_A = 1$

$J_B = \overline{Q}_D \cdot Q_A$ $\qquad$ $K_B = Q_A$

$J_C = Q_B \cdot Q_A$ $\qquad$ $K_C = Q_B \cdot Q_A$

$J_D = Q_C \cdot Q_B \cdot Q_A$ $\qquad$ $K_D = Q_A$

O circuito resultante é mostrado na figura 7.18:

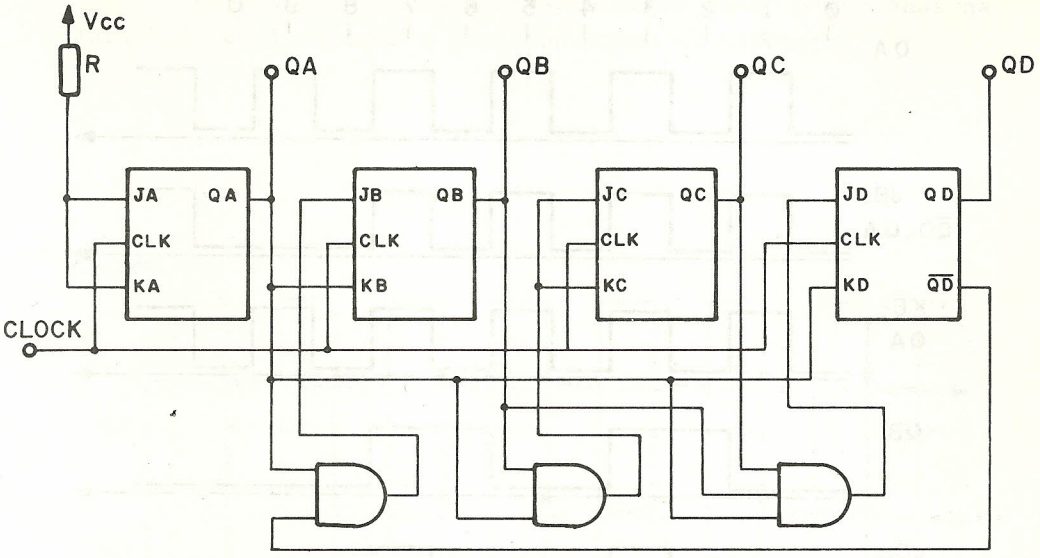


Fig. 7.18 - Contador Síncrono BCD 8421 com PARALLEL CARRY.

As formas de onda para este contador estão ilustradas na figura 7.19:

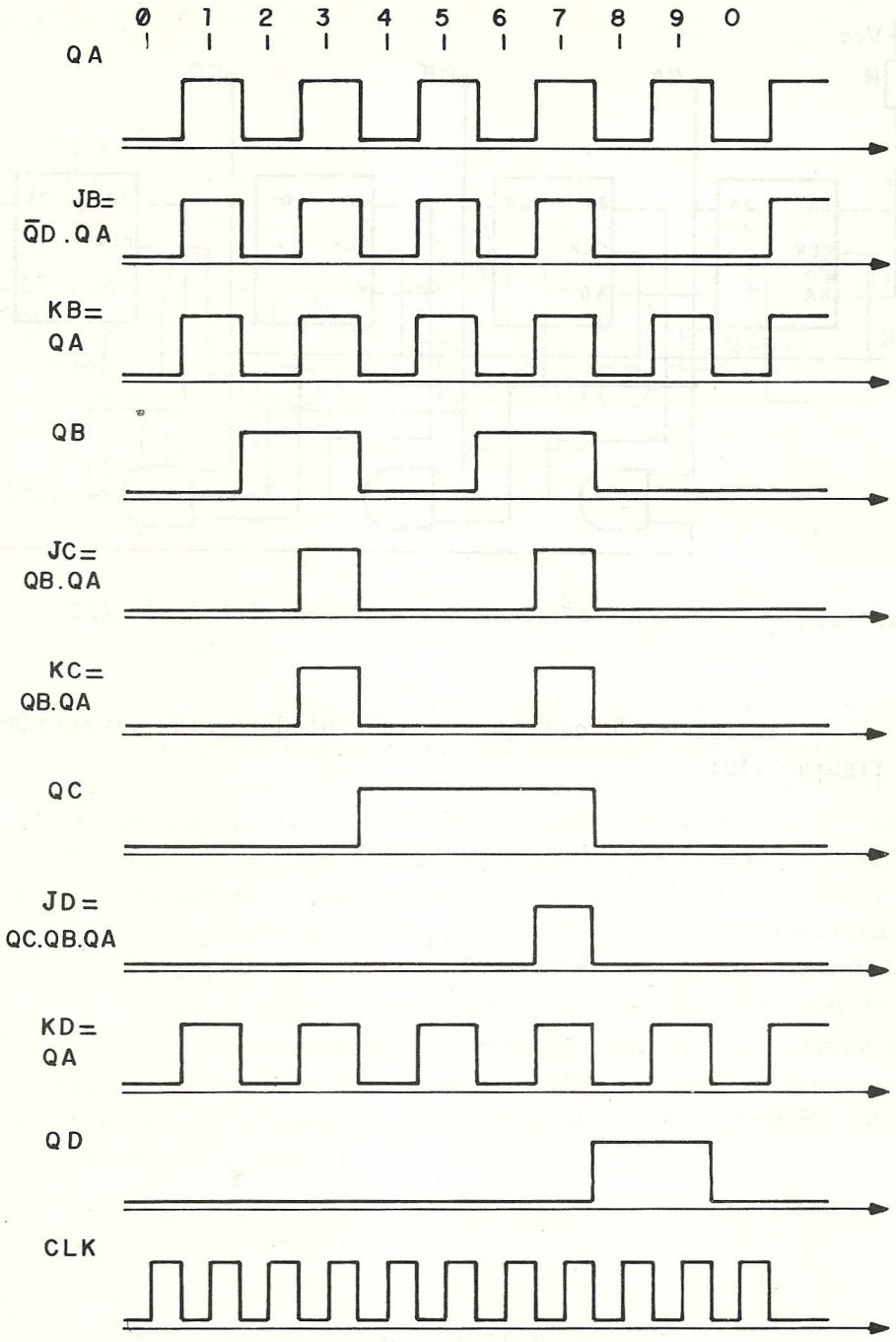


Fig. 7.19 - Formas de Onda para o Contador da figura 7.18.

Exemplo 7.4: Projetar um contador síncrono que conte segundo a tabela apresentada na figura 7.20:

Q_C	Q_B	Q_A	OBS:
0	0	1	1
0	1	1	3
1	0	1	5
1	1	1	7
1	1	0	6
1	0	0	4
0	1	0	2

Fig. 7.20 - Seqüência de contagem para o Contador do exemplo 7.4.

Solução:

Considerando a tabela de excitação do FF J-K construímos os mapas referentes às entradas J e K de cada FLIP-FLOP. Lembre-se: em cada estado verificamos como devem ser as entradas J e K para que ocorra a transição ao estado seguinte e plotamos nos mapas.

A figura 7.21 ilustra os seis mapas resultantes e as equações lógicas correspondentes:

QC \ QB QA	00	01	11	10
0	x	x	x	1
1	0	x	x	0

$J_A = \overline{Q}_C$

QC \ QB QA	00	01	11	10
0	x	0	0	x
1	x	0	1	x

$K_A = Q_C \cdot Q_B$

QC \ QB QA	00	01	11	10
0	x	1	x	x
1	1	1	x	x

$J_B = 1$

QC \ QB QA	00	01	11	10
0	x	x	1	1
1	x	x	0	1

$K_B = \overline{Q_C + Q_A} = Q_C \cdot Q_A$

QC \ QB QA	00	01	11	10
0	x	0	1	0
1	x	x	x	x

$J_C = Q_B \cdot Q_A$

QC \ QB QA	00	01	11	10
0	x	x	x	x
1	1	0	0	0

$K_C = \overline{Q}_B \cdot \overline{Q}_A$

Fig. 7.21 - Mapas de Karnaugh para o Contador do exemplo 7.4.

O circuito do contador está ilustrado na figura 7.22:

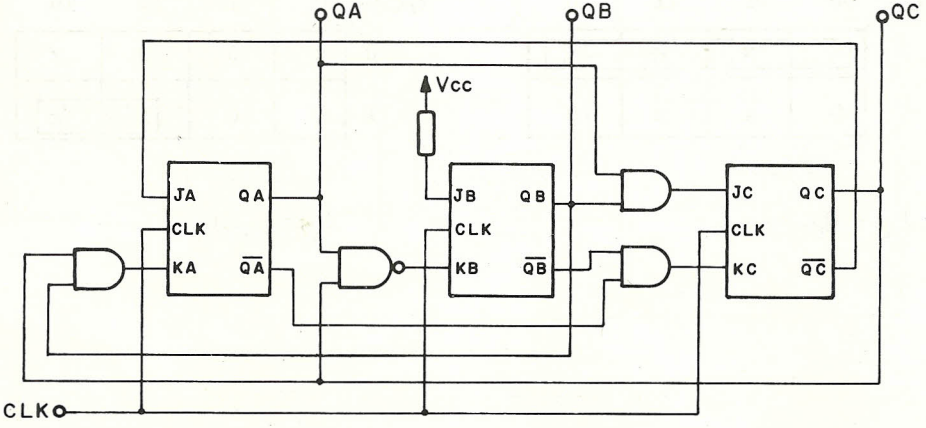


Fig. 7.22 - Circuito do Contador do exemplo 7.4.

Nos contadores dos exemplos 7.3 e 7.4 há estados que não fazem parte da seqüência de contagem. Quando o contador é ligado, pode acontecer que ele caia em um destes estados e é necessário analisar o que aconteceria em tal caso.

No contador da figura 7.22, por exemplo, vamos supor que os três FF's estivessem inicialmente ressetados. Neste caso, teríamos:

J_A = "1" J_B = "1" J_C = "Ø"

K_A = "Ø" K_B = "1" K_C = "1"

e o primeiro pulso de CLOCK faria com que o contador fosse para a contagem 3 (Q_A="1", Q_B="1", Q_C="Ø") e a partir daí ele entraria na seqüência normal, conforme ilustra a figura 7.23:

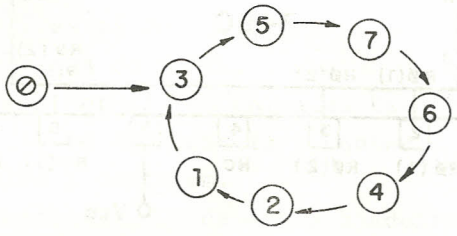


Fig. 7.23 - Seqüência de Estados para o Circuito da fig. 7.22

No entanto, pode ocorrer que o contador ao partir de um estado não pertencente à seqüência desejada fique preso em um laço e não entre na seqüência correta. Para evitar isto, é sempre recomendável fazer com que ele entre em um estado predeterminado, ao se aplicar a alimentação. Isto é conseguido ativando de maneira conveniente as linhas de PRESET e/ou CLEAR assíncronos no momento de ligação.

7.5 - Contadores da Família TTL

Nos itens anteriores, estudamos o projeto de contadores utilizando FLIP-FLOPS e portas lógicas. Em muitos casos, no entanto, é mais conveniente utilizar os contadores disponíveis na forma de circuitos integrados MSI, ou seja, um bloco funcional já completo. A seguir, apresentaremos alguns dos contadores disponíveis na série 74TTL, com suas características e aplicações.

7490 - DECADE COUNTER (÷ 10)

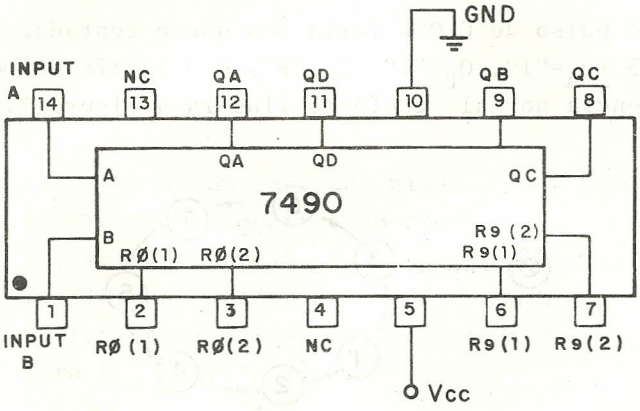


Fig. 7.24 - 7490.

Este CI contém quatro FF'S interconectados de modo a constituir um divisor por 2 e um divisor por 5. Os FF's são gatilhados na transição negativa do sinal de CLOCK. Os dois divisores podem ser usados independentemente, mas o RESET é comum a ambos. Se uma ou as duas entradas $R_Ø$ são colocadas em "1", o contador é ressetado para ØØØØ; se uma ou as duas entradas R_9 são colocadas em "1", o contador é pressetado para 1ØØ1.

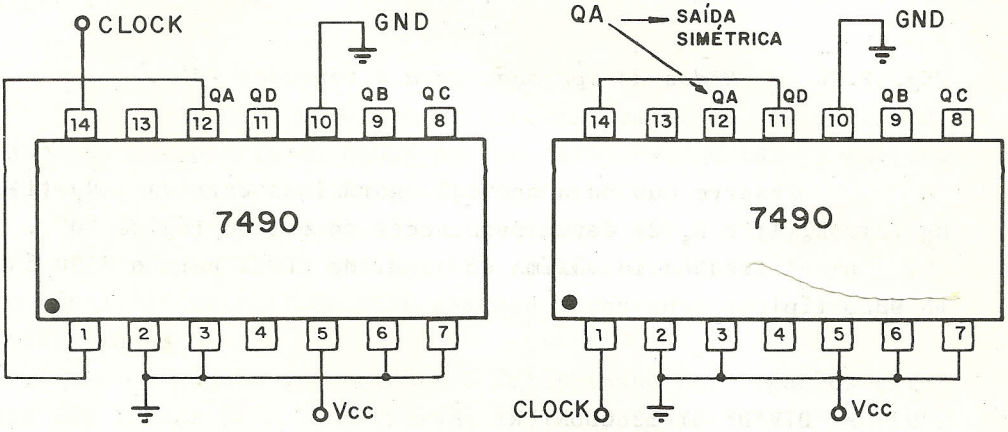
Há três modos de operação:

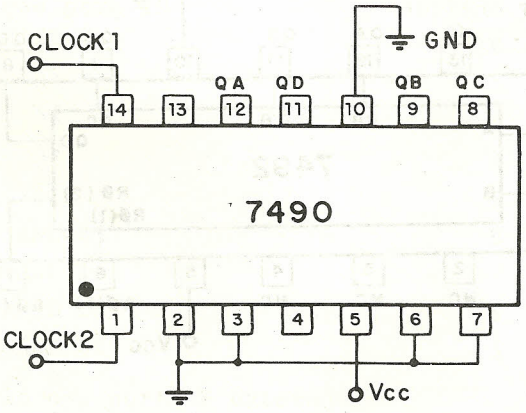
1) quando utilizado como um contador BCD, a entrada B deve ser externamente conectada à saída Q_A e o CLOCK aplicado à entrada A;

2) quando utilizado como um divisor por 1Ø simétrico, a saída Q_D deve ser externamente conectada à entrada A e o CLOCK aplicado à entrada B. A saída Q_A fornecerá uma onda quadrada com $^1/1Ø$ da freqüência do CLOCK;

3) quando utilizado como um divisor por 2 e um divisor por 5 independentes, não se requer nenhuma conexão externa. O FF A é o divisor por 2 e os FF's B, C e D formam o divisor por 5, cujo CLOCK é a entrada B.

A figura 7.25 mostra as três conexões possíveis e as tabelas da verdade correspondentes:





Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
0	0	0	1
0	1	1	1
0	1	0	1
0	1	1	1
1	0	0	1

Q_D	Q_C	Q_B
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

Fig. 7.25 - Modos de Operação para o Contador 7490.

Observe que para operação normal as entradas $R_{\emptyset}$ (1), $R_{\emptyset}$ (2), R_9(1) e R_9(2) devem permanecer no estado lógico "Ø" .

A freqüência máxima do sinal de CLOCK para o 7490 é de 18 MHZ, típico.

7492 - DIVIDE-BY-12 COUNTER

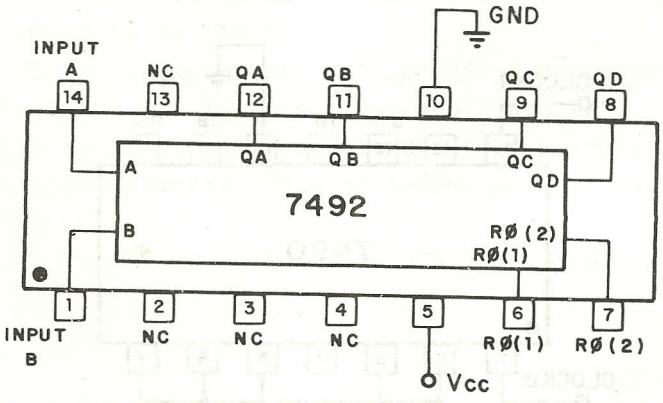


Fig. 7.26 - 7492.

Este CI contém quatro FF's interconectados de modo a fornecer um divisor por 2 e um divisor por 6, que podem ser usados independentemente. Os FF's são gatilhados na transição negativa do CLOCK. Se uma ou ambas entradas $R_{Ø}$ forem para "1", o contador será ressetado para ØØØØ.

Há três modos de operação:

1) se a saída Q_A for externamente conectada à entrada B, teremos um contador na seqüência ØØØØ, ØØØ1, ..., 1Ø11; o CLOCK deve ser aplicado à entrada A;

2) Se a saída Q_D for conectada externamente à entrada A, teremos um divisor por 12 com uma saída simétrica em Q_A; o CLOCK deve ser aplicado à entrada B;

3) se nenhuma ligação externa for feita, teremos um divisor por 2 e um divisor por 6 independentes, cujos CLOCKS são as entradas A e B, respectivamente.

A figura 7.27 mostra as tabelas da verdade para os três modos de operação:

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1

divisor por 12

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	0
1	0	1	0
1	1	0	0
0	0	0	1
0	0	1	1
0	1	0	1
1	0	0	1
1	0	1	1
1	1	0	1

divisor por 12

Q_D	Q_C	Q_B
0	0	0
0	0	1
0	1	0
1	0	0
1	0	1
1	1	0

divisor por 6

Fig. 7.27 - Modos de Operação para o Contador 7492.

Para operação normal, as entradas $R_{\emptyset}$ (1) e $R_{\emptyset}$ (2) devem permanecer no estado "Ø". A freqüência máxima do CLOCK é 18 MHZ, tipicamente.

7493 - 4 - BIT BYNARY COUNTER

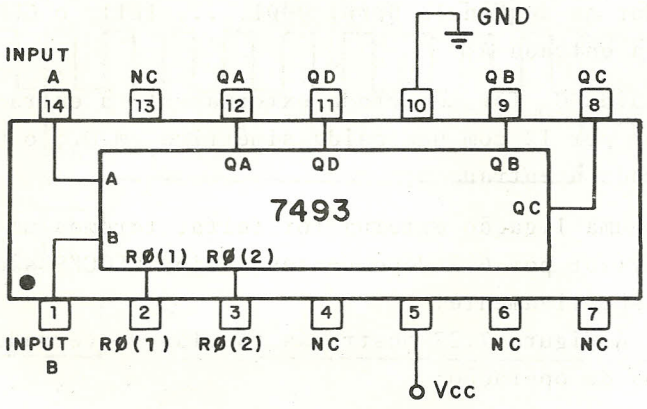


Fig. 7.28 - 7493.

Este CI contém quatro FLIP-FLOPS conectados de modo a fornecer um divisor por 2 e um divisor por 8, que podem ser usados independentemente. Os FF's são gatilhados na transição negativa do CLOCK. Se uma ou as duas entradas $R_{\emptyset}$ forem para "1", o contador será ressetado para ØØØØ.

Há dois modos de operação:

1) se a saída Q_A for aplicada à entrada B, o contador irá contar na seqüência ØØØØ, ØØØ1, ..., 1111 e teremos nas saídas Q_A, Q_B, Q_C e Q_D divisões por 2, 4, 8 e 16, respectivamente. O CLOCK será aplicado na entrada A;

2) se nenhuma conexão externa for feita, teremos um divisor por 2 e um divisor por 8 independentes, cujos CLOCKS são as entradas A e B, respectivamente.

A figura 7.29 mostra as tabelas da verdade para os dois modos de operação:

Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

divisor por 16

Q_D	Q_C	Q_B
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

divisor por 8

Fig. 7.29 - Modos de Operação para o Contador 7493.

Para operação normal as entradas $R_{\emptyset}$ (1) e $R_{\emptyset}$ (2) devem permanecer em "Ø". A freqüência máxima do CLOCK é de 18 MHZ, tipicamente.

74161 - BINARY (÷ 16) COUNTER (SYNCHRONOUS,PRESETTABLE,UNIT - CASCADABLE)

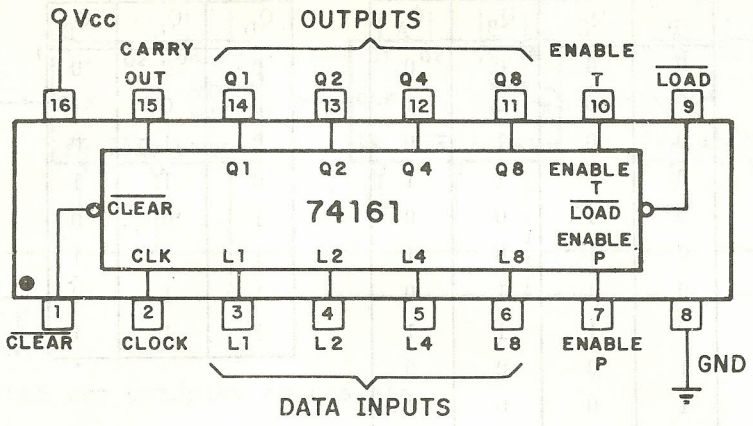


Fig. 7.30 - 74161.

Este CI é um contador binário síncrono de 4 bits, gatilhado na transição positiva do CLOCK. Para operação normal as entradas ENABLE P, ENABLE T, $\overline{\text{CLEAR}}$ e $\overline{\text{LOAD}}$ devem ficar no estado "1". Se uma ou as duas entradas de ENABLE estiverem em "Ø", o contador não avançará. Um nível "Ø" na entrada $\overline{\text{CLEAR}}$ provoca o RESET assíncrono do contador. Se a entrada $\overline{\text{LOAD}}$ estiver baixa, o conteúdo das entradas L_1, L_2, L_4 e L_8 será transferido para as saídas Q_1, Q_2, Q_4 e Q_8 na transição positiva do CLOCK.

A saída CARRY OUT fixará em "1" quando a contagem for igual a 1111. Vários estágios podem ser ligados em cascata conectando-se a saída CARRY OUT de um estágio à entrada ENABLE T do estágio seguinte.

A freqüência máxima do sinal de CLOCK é de 25 MHZ, tipicamente.

74191 - BYNARY (÷ 16) UP/DOWN COUNTER (SYNCHRONOUS, PRESETTABLE).

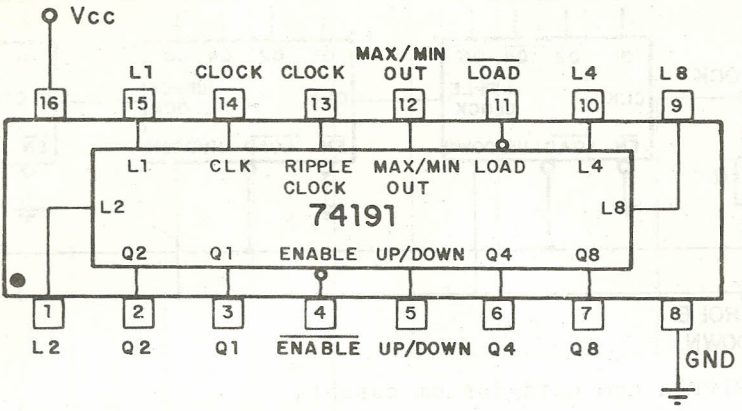


Fig. 7.31 - 74191.

Este CI é um contador binário síncrono progressivo/regressivo de quatro bits, gatilhado na transição positiva do CLOCK.

Em operação normal a entrada $\overline{\text{ENABLE}}$ deve ser "Ø" e a entrada $\overline{\text{LOAD}}$ deve ser "1". Um nível "1" em $\overline{\text{ENABLE}}$ inibe a contagem, um nível "Ø" em $\overline{\text{LOAD}}$ faz com que o conteúdo das entradas L_1, L_2, L_4 e L_8 seja transferido assincronamente para a saída.

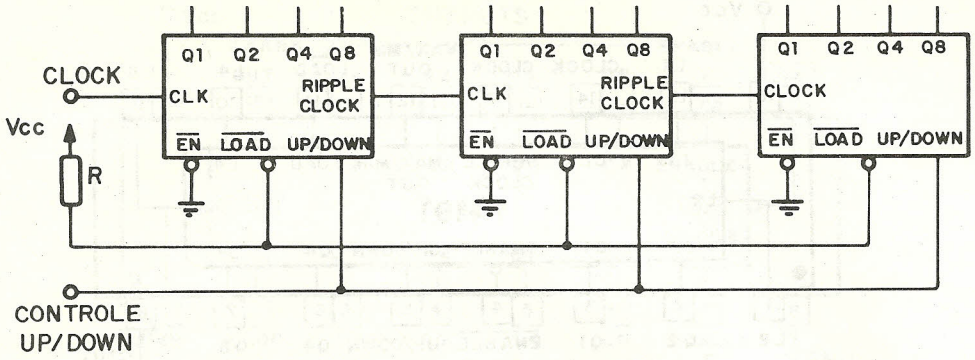
Se a entrada UP/DOWN está em "Ø", a contagem é progressiva; se está em "1", a contagem é regressiva. A saída MAX/MIN fica em "1" quando:

a) a contagem é progressiva e o contador contém 1111;

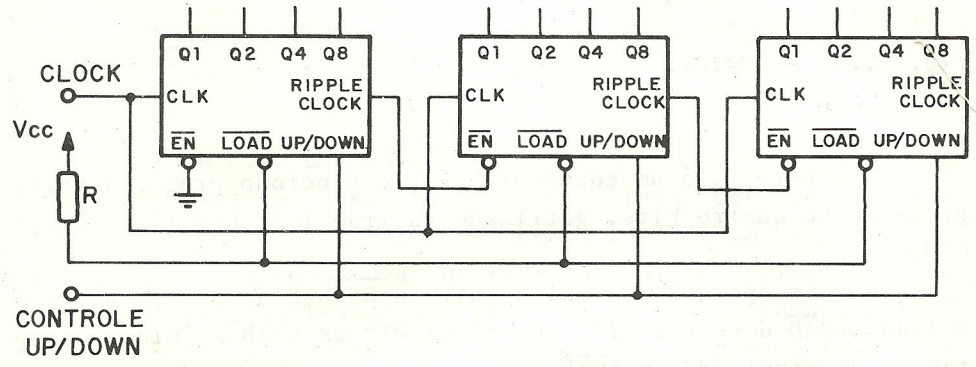
b) a contagem é regressiva e o contador contém ØØØØ.

O contador não tem a linha de CLEAR, que é conseguido carregando-o com ØØØØ.

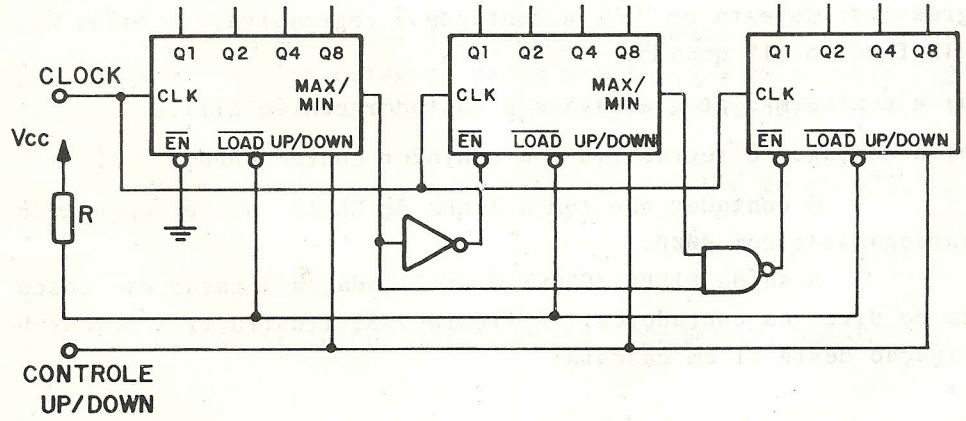
A saída RIPPLE/CLOCK é utilizada na ligação em cascata de diversos contadores. A figura 7.32 ilustra três formas de ligação deste CI em cascata:



(a) RIPPLE com estágios em cascata



(b) Contador síncrono com RIPPLE CARRY



(c) Contador síncrono com PARALLEL CARRY

Fig. 7.32 - Conexão do 74191 em cascata.

As três ligações mostradas na figura 7.32 correspondem àquelas já apresentadas para os contadores construídos com FLIP-FLOPS. A ligação C é a que permite a maior freqüência do sinal de CLOCK.

A freqüência máxima do sinal de CLOCK para o CI 74191 é tipicamente de 20 MHZ.

7.6 - Contadores da Família C-MOS

Também na família C-MOS dispomos de uma série de contadores MSI que, em muitos casos, são preferíveis à utilização de FLIP-FLOPS. A seguir, apresentaremos alguns contadores da série 4ØØØ C-MOS com suas características principais.

4017 - DECADE COUNTER/DIVIDER

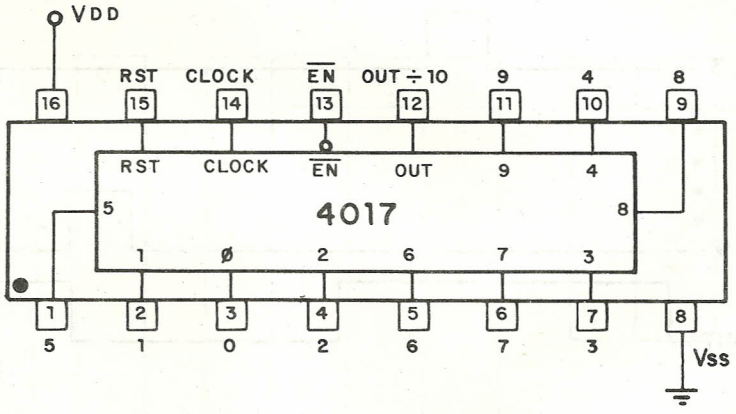


Fig. 7.33 - 4017.

Este CI é uma década contadora síncrona com as saídas já decodificadas. O gatilhamento ocorre na transição positiva do CLOCK.

Para operação normal, as linhas RST e $\overline{EN}$ devem ficar em "Ø". Se a entrada RST vai para "1", o contador é ressetado para "Ø"; se a entrada $\overline{EN}$ vai para "1", a contagem é inibida.

A cada transição positiva do CLOCK a saída correspondente à contagem presente vai para "1", permanecendo as demais em "Ø". A saída OUT fornece uma onda quadrada com 1/10 da freqüência do sinal de CLOCK. A figura 7.34 ilustra as formas de onda para este contador:

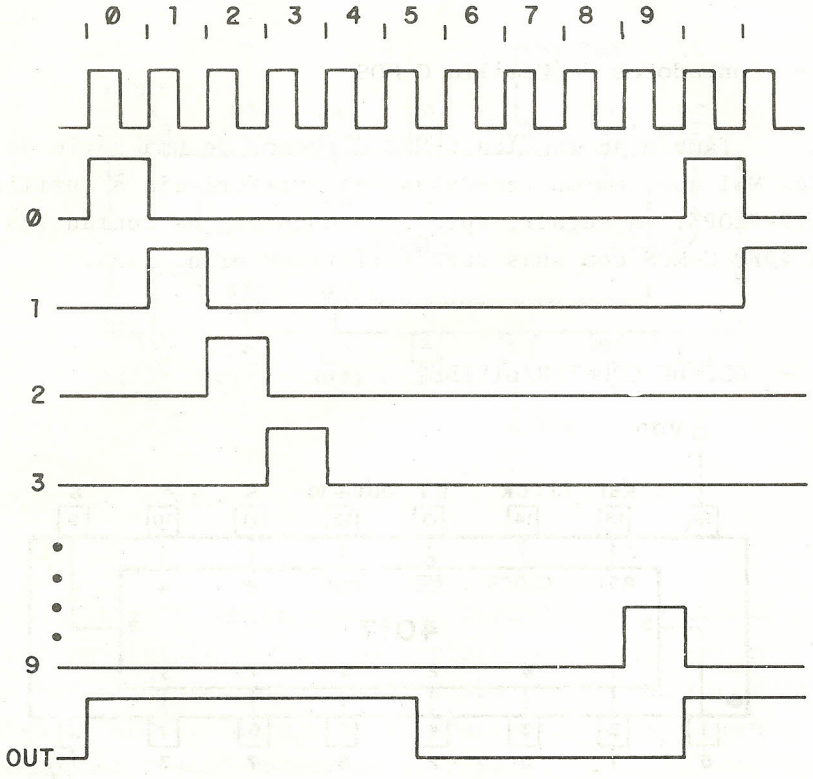


Fig. 7.34 - Formas de onda para o Contador 4017.

A freqüência máxima do CLOCK é de 5 MHZ para V_{cc} = 10V e 2,5 MHZ para V_{cc} = 5V.

4018 - PRESETTABLE DIVIDE-BY-N COUNTER

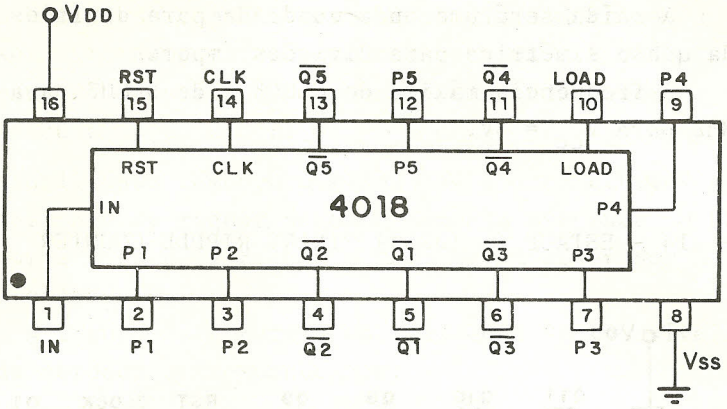


Fig. 7.35 - 4018.

Este CI é um contador síncrono capaz de efetuar a divisão da freqüência do CLOCK por um valor de 2 a 1Ø. O gatilhamento ocorre na transição positiva do CLOCK.

Em operação normal, as entradas RST e LOAD ficam em "Ø". Um nível "1" na entrada RST resseta o contador para "Ø", um nível "1" na entrada LOAD carrega o contador com o conteúdo das entradas P.

O módulo do contador é determinado por ligações externas das saídas $\overline{Ø1}$ a $\overline{Ø5}$ à entrada IN, conforme a tabela apresentada na figura 7.36:

Ligação para IN	Divisão por
$\overline{05}$	10
$\overline{04}$	8
$\overline{03}$	6
$\overline{02}$	4
$\overline{01}$	2
* $\overline{04}$ AND $\overline{05}$	9
* $\overline{03}$ AND $\overline{04}$	7
* $\overline{02}$ AND $\overline{03}$	5
* $\overline{01}$ AND $\overline{02}$	3

* - é necessário uma porta E externa

Fig. 7.36 - Conexões Externas para o CI 7418.

A saída será uma onda quadrada para divisões pares e uma onda quase simétrica para divisões ímpares.

A freqüência máxima do CLOCK é de 5 MHZ para V_{cc} = 10V e 2,5 MHZ para V_{cc} = 5V.

4020 - 14 - ESTAGE (÷ 16384) BINARY RIPPLE COUNTER

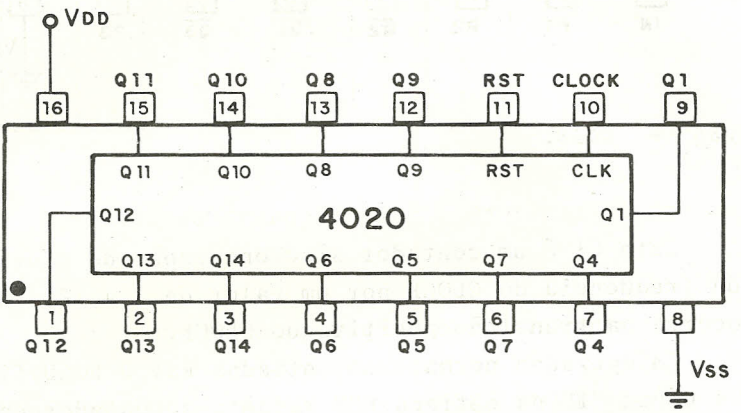


Fig. 7.37 - 4020.

Este CI é um contador binário de 14 bits, gatilhado na transição negativa do CLOCK.

Em operação normal, a linha RST deve ficar em "Ø"; um nível "1" nesta linha resseta o contador para "Ø".

Na saída Q_1 a freqüência do CLOCK aparece dividida por 8 e assim sucessivamente. Não há saídas para os estágios 2 e 3.

A freqüência máxima do CLOCK é de 7 MHZ em V_{cc} = 10 V e 2,5 MHZ para V_{cc} = 5 V.

4022 - DIVIDE-BY-8 COUNTER WITH 1-OF-8 OUTPUTS (SYNCHRONOUS)

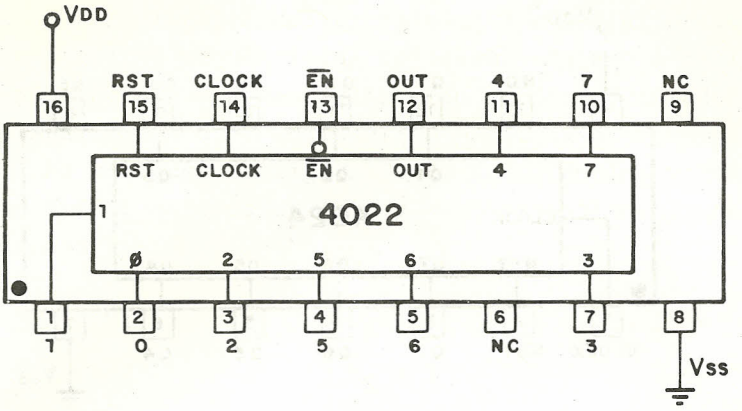


Fig. 7.38 - 4022.

Este CI é similar ao 4017, constituindo-se em um divisor por 8 com as saídas já decodificadas. O contador é gatilhado na transição positiva do CLOCK.

Em operação normal, as entradas $\overline{EN}$ e RST permanecem em "Ø". Um nível "1" em RST resseta o contador para "Ø" e um nível "1" em $\overline{EN}$ inibe a contagem.

A cada contagem a saída correspondente irá para "1", permanecendo as demais em "Ø". A saída OUT fornece uma onda quadrada com $1/8$ da freqüência do CLOCK, sendo "1" para as contagens Ø, 1, 2, 3 e Ø para as contagens 4, 5, 6 e 7.

A freqüência máxima do CLOCK é de 5 MHZ para V_{cc} = 10V e 2,5 MHZ para V_{cc} = 5V.

4024 - 7-STAGE (÷ 128) BINARY RIPPLE COUNTER

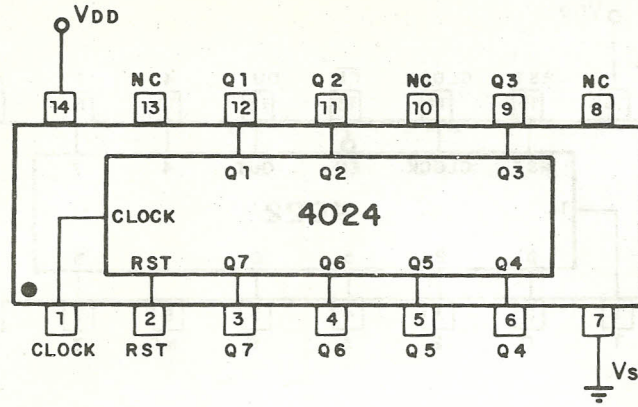


Fig. 7.39 - 4024.

Trata-se de um contador binário assíncrono de 7 bits, gatilhado na transição negativa do clock.

Em operação normal, a linha RST permanece em "Ø"; se RST vai para "1", o contador é ressetado para a contagem "Ø".

A freqüência máxima do CLOCK é de 7 MHZ para V_{cc} = 10V e 2,5 MHZ para V_{cc} = 5V.

4029 - UP-DOWN DIVIDE-BY-10 OR DIVIDE-BY-16 COUNTER (SYNCHRONOUS)

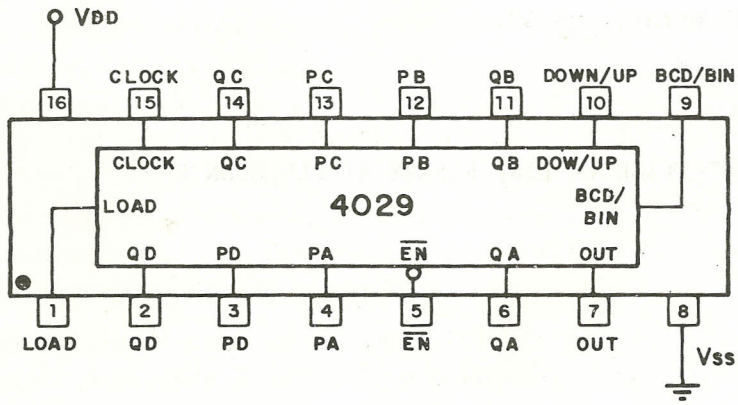


Fig. 7.40 - 4029.

Este CI contém um contador síncrono que pode operar como uma década ou como um contador binário de 4 bits. Se a entrada BCD/BIN está em "Ø" , o contador se comporta como uma década (divisor por 10); se BCD/BIN está em "1", se comporta como um divisor por 16. Em qualquer dos casos, a contagem pode ser regressiva (DOWN/UP = "Ø") ou progressiva (DOWN/UP = "1"). O gatilhamento ocorre na transição positiva do CLOCK.

Para operação normal, as entradas LOAD e $\overline{EN}$ devem ficar em "Ø". Se $\overline{EN}$ fica em "1", a contagem é inibida, se LOAD fica em "1", o conteúdo das entradas P_D, P_C, P_B e P_A é transferido para as saídas. O contador é ressetado carregando-se o valor ØØØØ através das entradas de dados.

A saída OUT é ativa no estado "Ø", ou seja, permanece em "1" na maior parte do tempo, indo para "Ø" quando a contagem é 9 e BCD/BIN = "Ø" ou a contagem é F e BCD/BIN = "1". Esta saída é ligada à entrada $\overline{EN}$ do estágio seguinte para a ligação em cascata de vários contadores.

A freqüência máxima do CLOCK é de 5 MHZ para V_{CC} = 10V e 2,5 MHZ para V_{CC} = 5V.

7.7 - Aplicações de Contadores MSI

Em itens anteriores, vimos a construção de contadores síncronos e assíncronos utilizando FLIP-FLOPS. Através do uso das linhas de CLEAR e/ou LOAD podemos também construir contadores de módulo menor que o máximo, ou seja, diminuir o ciclo de contagem, utilizando os blocos MSI apresentados nos itens 7.5 e 7.6.

A figura 7.41 mostra, como exemplo, um divisor por 6 construído com um 7490. Neste circuito quando a contagem 6 é atingida (Ø11Ø), as linhas de CLEAR assíncrono são ativadas e o contador é ressetado para "Ø".

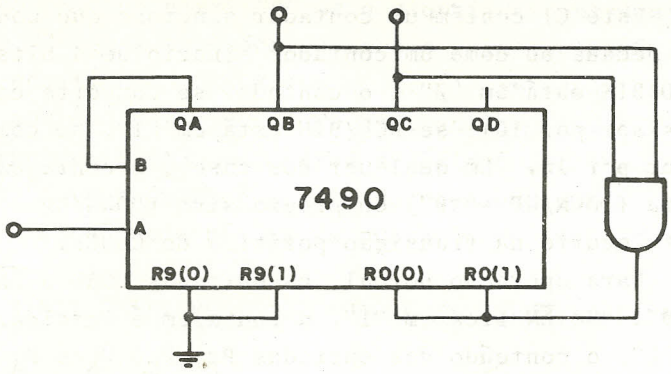


Fig. 7.41 - Divisor por 6 com CI 7490.

A contagem 6 é instável e dura apenas alguns nanosegundos, conforme ilustram a tabela da verdade e as formas de onda apresentadas na figura 7.42:

	Q_C	Q_B	Q_A
tn	0	0	0
tn+1	0	0	1
tn+2	0	1	0
tn+3	0	1	1
tn+4	1	0	0
tn+5	1	0	1
tn+6	1/0	1/0	0/0

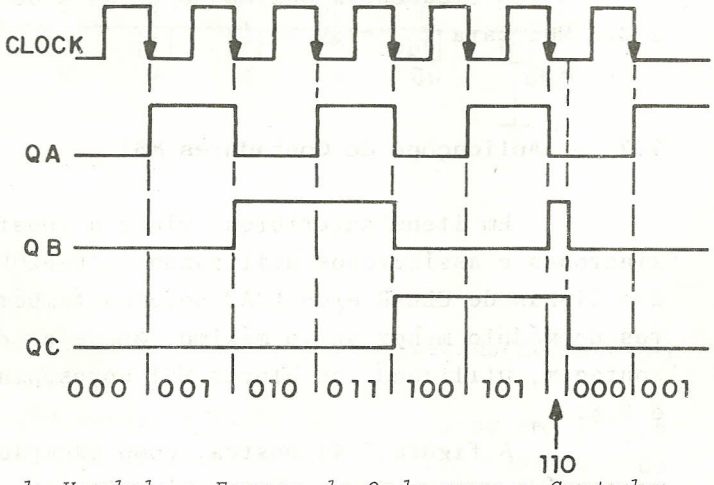


Fig.7.42 - Tabela da Verdade e Formas de Onda para o Contador da figura 7.41.

Um contador de módulo variável pode ser construído utilizando CI's com capacidade de carregamento paralelo. A figura 7.43 mostra um contador divisor por N utilizando o CI 74161. O módulo deste contador será:

N = 255 - (conteúdo das entradas paralelas)

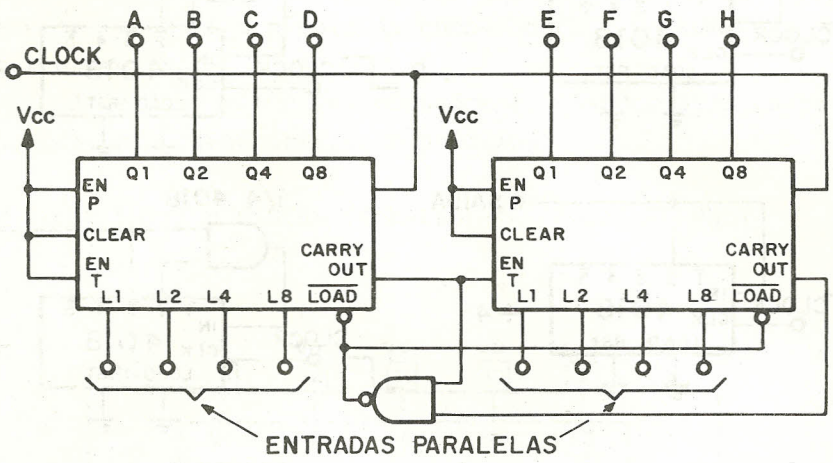


Fig. 7.43 - Contador Síncrono de Módulo variável com 74161.

O circuito consiste de dois estágios ligados em cascata, sendo o segundo habilitado pela saída CARRY-OUT do primeiro. Quando os dois estágios atingem a contagem máxima (F) as entradas $\overline{\text{LOAD}}$ são habilitadas e na próxima transição positiva do sinal de CLOCK o conteúdo das entradas paralelas é carregado no contador. Assim, o contador terá um módulo igual à diferença entre o valor carregado nas entradas e a contagem máxima FF(= 255). Na saída da porta NAND, teremos um pulso a cada N pulsos de CLOCK, sendo N o módulo.

O CI 4018, pode ser utilizado como um divisor de módulo variável de 2 a 10, dependendo de conexões externas. A figura 7.44 ilustra as possíveis ligações deste CI.

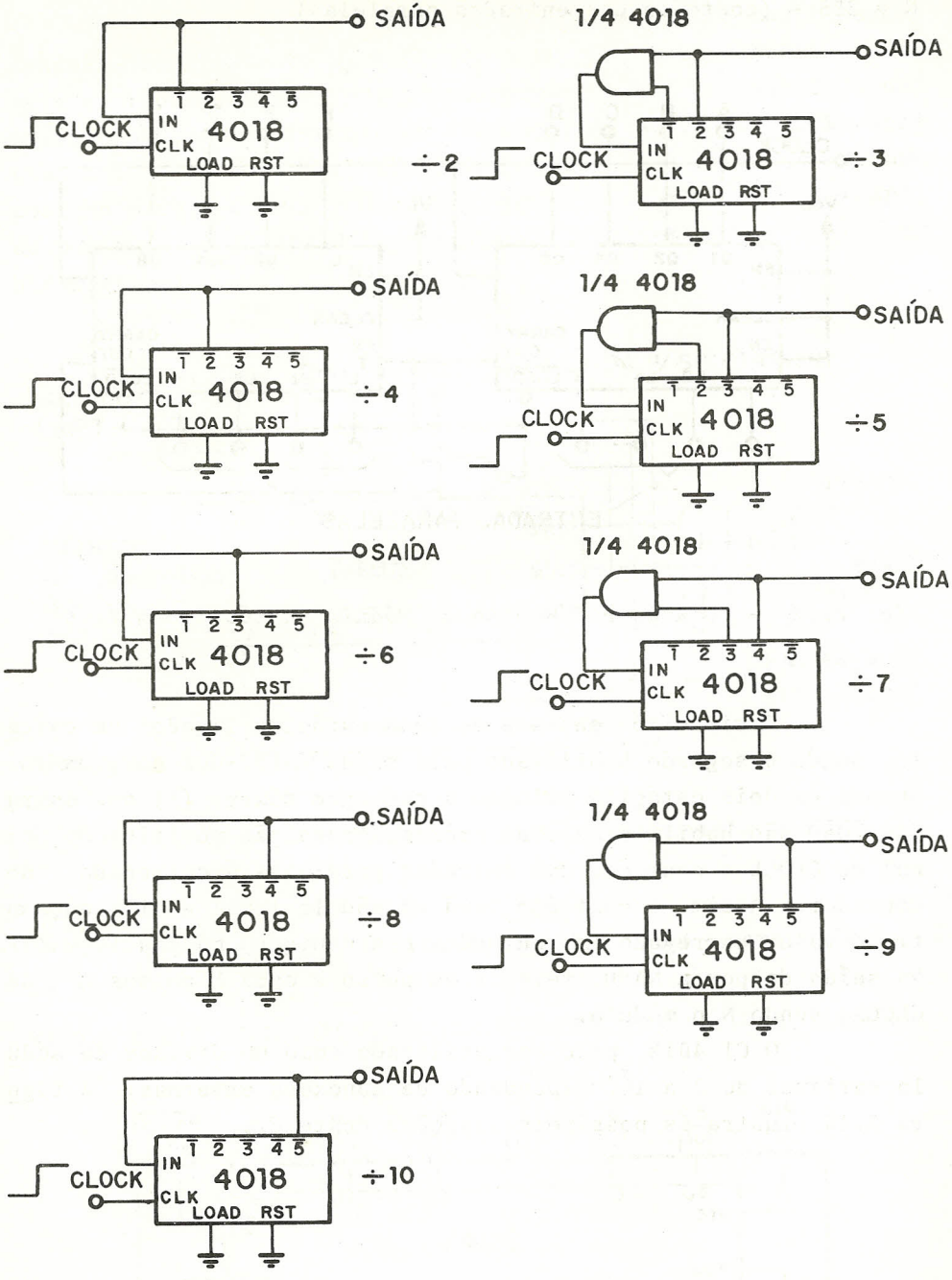


Fig. 7.44 - Conexões do CI 4018 como divisor.

Os contadores digitais têm grande aplicação em instrumentação eletrônica. Um exemplo interessante é um freqüencímetro digital. Neste aparelho, a medida de freqüência é feita contando-se o número de pulsos do sinal que se quer medir durante um intervalo de tempo bem definido, digamos 1 segundo. A figura 7.45 mostra o diagrama em blocos da parte de contagem de um freqüencímetro:

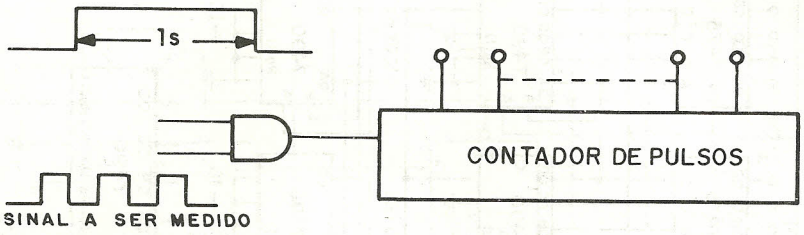


Fig. 7.45 - Princípio de Medição de Freqüência.

O sinal de habilitação para a porta lógica denomina-se base de tempo e é, em geral, derivado de um oscilador a cristal de grande estabilidade, através de sucessivas divisões de freqüência. Partindo de um oscilador de 10 MHZ, por exemplo, e dividindo a freqüência por 10^7 teríamos um sinal de período igual a 1s. Efetuando mais uma divisão, desta vez por 2, obteríamos um sinal de freqüência 0,5 HZ, que ficaria exatamente 1s no estado lógico "1" e um segundo no estado "Ø".

A figura 7.46 mostra o esquema básico da parte de geração de base de tempo e contagem de um freqüencímetro digital, utilizando CI's TTL.

para o decodificador e DISPLAY

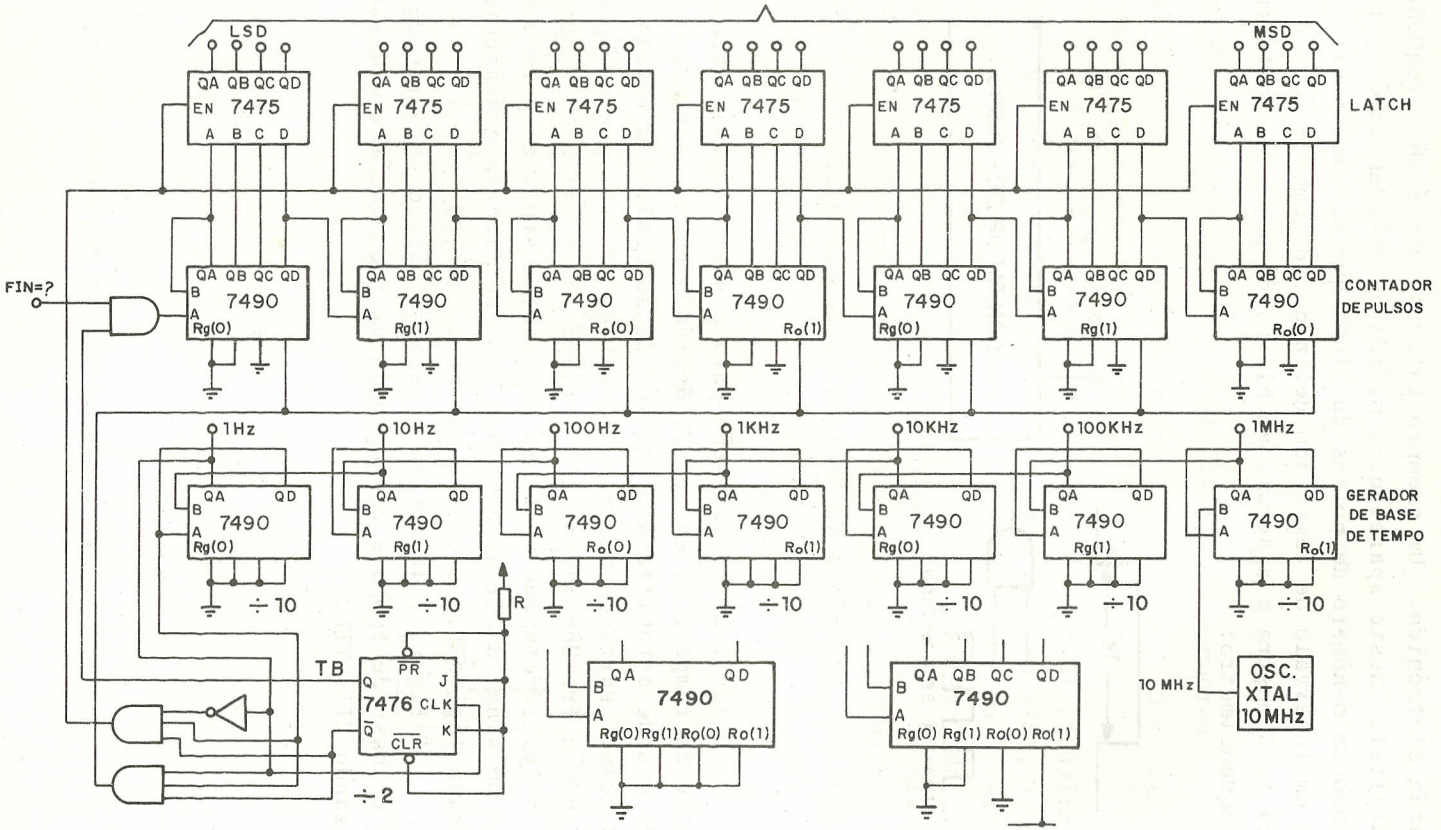


Fig. 7.46 - Geração de Base de Tempo e Contador de Pulsos para freqüencímetro digital.

O gerador de base de tempo parte de um oscilador a cristal de 10 MHZ cuja freqüência é dividida sucessivamente por contadores 7490 ligados em cascata. Após 7 divisões por 10, o sinal de 1 HZ resultante é dividido por 2 através de um FF JK ligado como TOGGLE. Observe que os CI's 7490 estão conectados de maneira a efetuar a divisão simétrica por 10.

No contador de pulsos temos sete CI's 7490 interligados de maneira a efetuar a contagem dos pulsos em BCD (décadas contadoras). A saída do contador de pulsos é transferida para um LATCH, constituído de sete CI's 7475. Esta transferência ocorre quando o contador de pulsos terminou a contagem e não está recebendo pulsos na entrada. Após a transferência do resultado para o LATCH o contador é ressetado para "Ø" e fica preparado para iniciar novamente a contagem. A figura 7.47 ilustra as formas de onda envolvidas no controle do circuito da figura 7.46. A saída do LATCH é levada ao circuito de decodificação e DISPLAY não ilustrado na figura. A finalidade do LATCH é evitar que o display mostre a contagem, o que ocorreria se a saída do contador de pulsos fosse levada diretamente ao circuito de decodificação e DISPLAY.

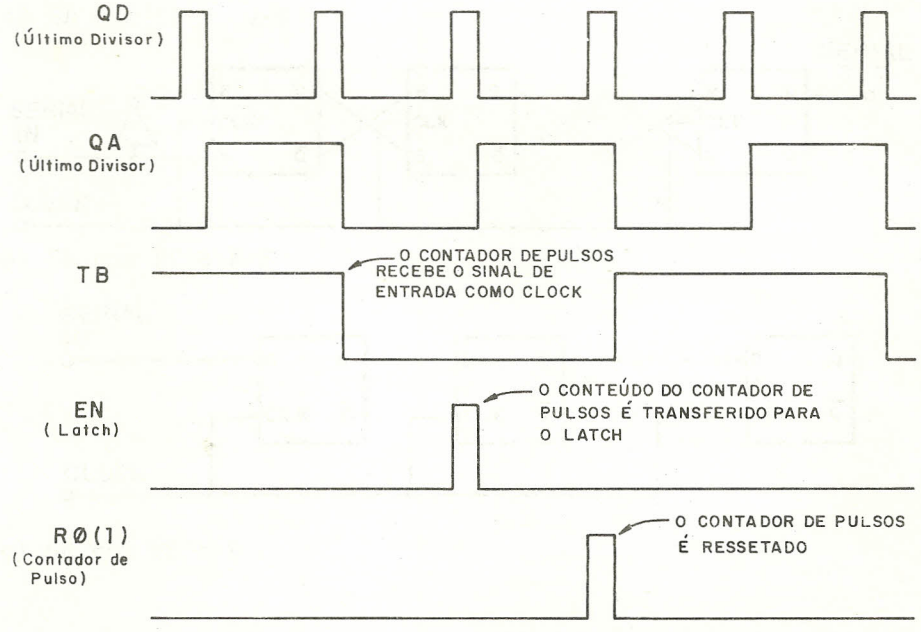


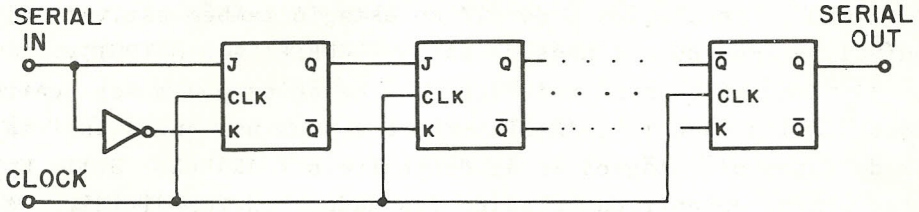
Fig. 7.47 - Formas de Onda de controle para o circuito da figura 7.46.

CAPÍTULO 8 - REGISTRADORES DE DESLOCAMENTO

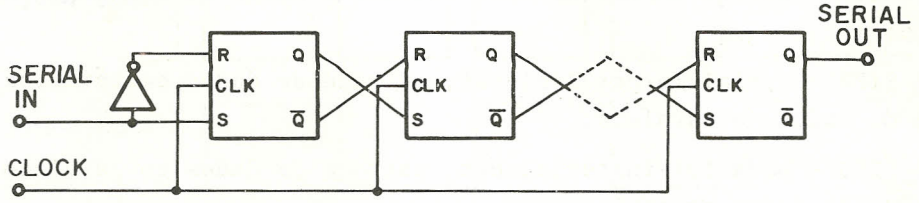
8.1 - Introdução

Um registrador de deslocamento ou "SHIFT REGISTER" é um circuito constituído de FLIP-FLOPS no qual a informação é deslocada de uma posição a cada pulso de CLOCK. Existem registradores de deslocamento (SR's) capazes de efetuar deslocamentos para a direita para a esquerda ou nas duas direções.

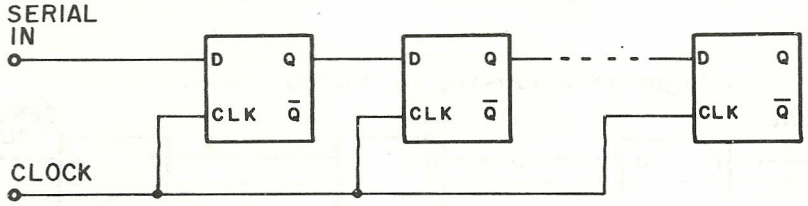
A figura 8.1 ilustra o diagrama básico dos SR's construídos com FF's J-K, R-S e D:



(a) SR com FF's J-K



(b) SR com FF's R-S



(c) SR com FF's D

Fig. 8.1 - Registradores de Deslocamento Básico.

Os registradores de deslocamento encontram grande aplicação na prática em conversores série/paralelo e paralelo/série, contadores, geradores de palavra, multiplicadores binários, memórias, etc.

8.2 - Tipos de SR's

No SR mais simples, como os da figura 8.1, temos apenas uma entrada de dados serial (SINGLE RAIL INPUT) e uma saída de dados serial (SINGLE RAIL OUTPUT). Se no circuito da figura 8.1(a) for retirado o inversor entre as entradas J e K do primeiro estágio, obteremos um SR com duas entradas seriais (DOUBLE RAIL INPUT); se a saída $\overline{Q}$ do último estágio também estiver disponível teremos duas linhas de saída (DOUBLE RAIL OUTPUT).

Há SR's nos quais todos os estágios podem ser carregados simultaneamente (PARALLEL-IN) e outros nos quais as saídas de todos os estágios estão disponíveis (PARALLEL-OUT). Conforme o modo de entrada e saída dos dados podemos dividir os SR's em quatro categorias.

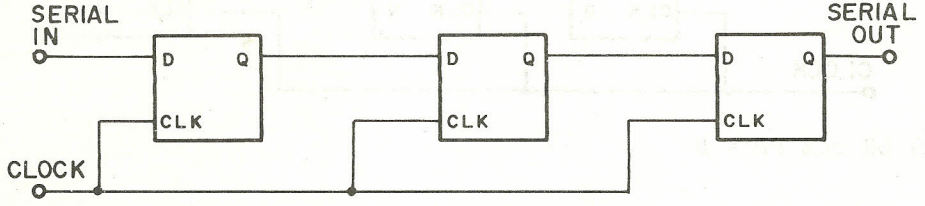
a) SISO - serial-in/serial-out, entrada e saída de dados em série;

b) SIPO - serial-in/parallel-out, entrada de dados em série, saída de dados em paralelo;

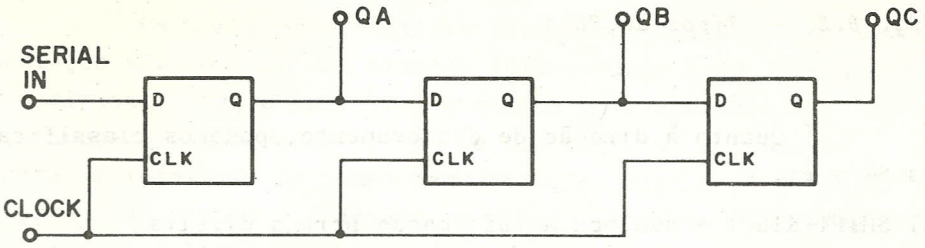
c) PISO - parallel-in/serial-out, entrada de dados em paralelo, saída de dados em série;

d) PIPO - parallel-in/parallel-out, entrada e saída de dados em paralelo.

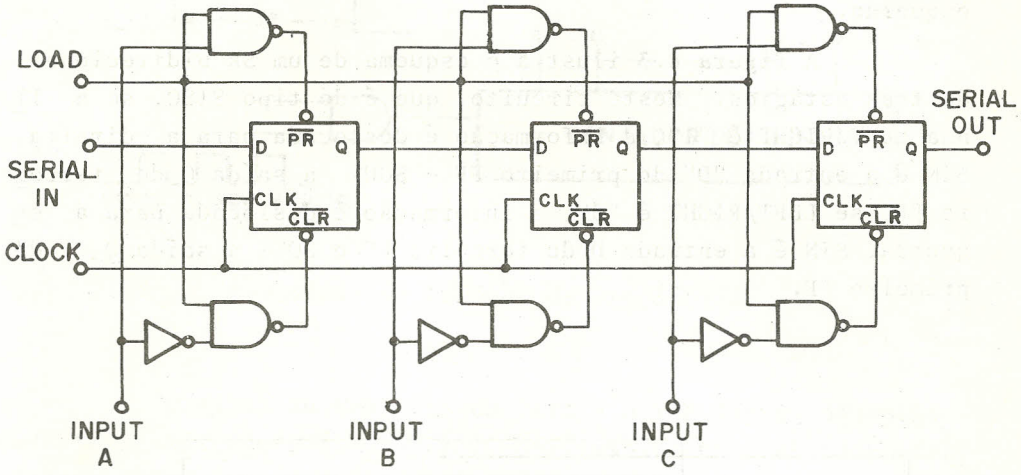
A figura 8.2 ilustra os quatro tipos:



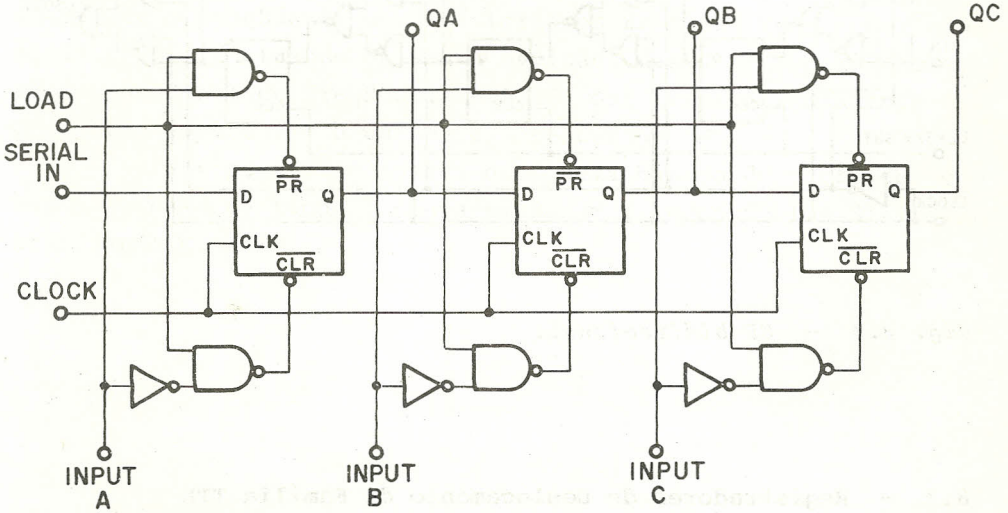
(a) SISO - SERIAL-IN/SERIAL-OUT



(b) SIPO - SERIAL-IN/PARALLEL-OUT



(c) PISO - PARALLEL-IN/SERIAL-OUT



(d) PIPO - PARALLEL-IN/PARALLEL-OUT

Fig. 8.2 - Tipos de SR's.

Quanto à direção de deslocamento, podemos classificar os SR's em:

a) SHIFT-RIGHT - desloca a informação para a direita;

b) SHIFT-LEFT - desloca a informação para a esquerda;

c) BIDIRECIONAL - desloca a informação para a direita ou para a esquerda.

A figura 8.3 ilustra o esquema de um SR bidirecional, de três estágios. Neste circuito, que é do tipo SISO, se a linha LEFT/RIGHT é "1", a informação é deslocada para a direita, SIN é a entrada "D" do primeiro FF e SOUT a saída Q do terceiro FF; se LEFT/RIGHT é "Ø", a informação é deslocada para a esquerda, SIN é a entrada D do terceiro FF e SOUT a saída Q do primeiro FF.

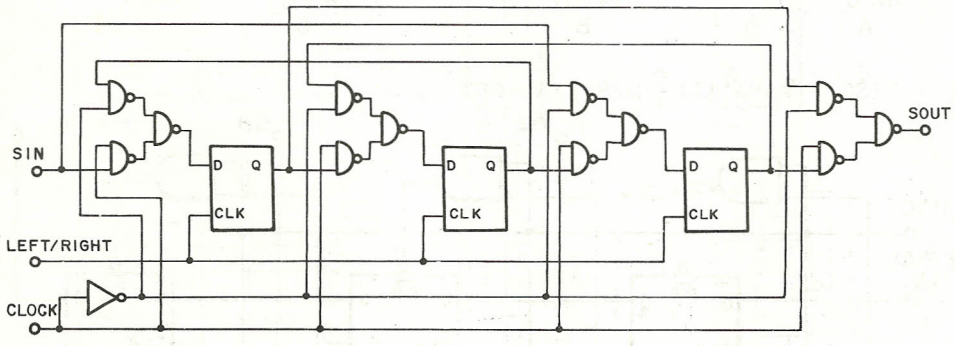


Fig. 8.3 - SR bidirecional.

8.3 - Registradores de Deslocamento da Família TTL

No item anterior, ilustramos alguns SR's construídos com FLIP-FLOPS. Na prática é mais conveniente utilizarmos os circuitos integrados MSI disponíveis nas famílias TTL e CMOS, que permitem reduzir o número de componentes necessários e simplificar o projeto.

A seguir, estão relacionados alguns dos SR's da série 74 TTL, com suas características principais.

7495 - 4-BIT SHIFT REGISTER (PARALLEL-IN, PARALLEL OUT)

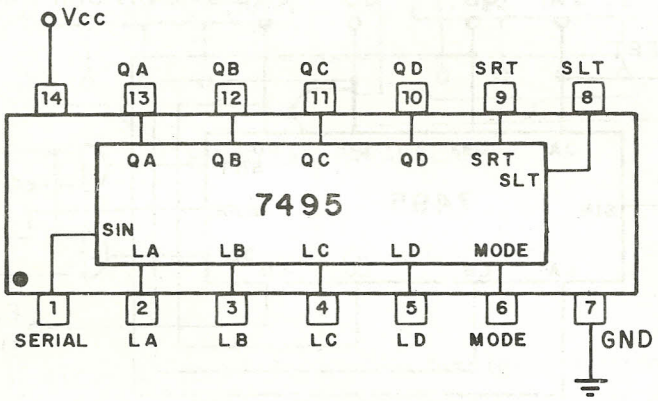


Fig. 8.4 - 7495.

Este CI contém um SR de 4 bits com entrada e saída de dados em paralelo. Dependendo da maneira de conexão, pode ser feito o deslocamento para a direita ou para a esquerda.

Quando se deseja utilizá-lo como um SR para a direita apenas, a entrada SLT deve ficar em "Ø". Neste caso, um nível "1" na entrada MODE carrega assincronamente as entradas LA, LB, LC e LD no registrador. Com MODE = "Ø" os dados serão deslocados de uma posição para a direita a cada transição negativa do CLOCK, que deve ser aplicado na entrada SRT. O deslocamento consiste em transferir o conteúdo de SIN para Q_A, Q_A para Q_B, Q_B para Q_C e Q_C para Q_D; o conteúdo do último estágio é destruído ou transferido para o estágio seguinte, caso diversos CI's estejam ligados em cascata.

O CI pode ser convertido em um SR bidirecional conectando-se externamente Q_D a LC, Q_C a LB e Q_B a LA. Neste caso, se a entrada MODE for "Ø", a informação será deslocada de uma posição para a direita a cada transição negativa do CLOCK aplicado a SRT; se MODE for "1", a informação será deslocada de uma posição para a esquerda a cada transição negativa do CLOCK aplicado a SLT. A figura 8.5 mostra a conexão deste CI como um SR bidirecional.

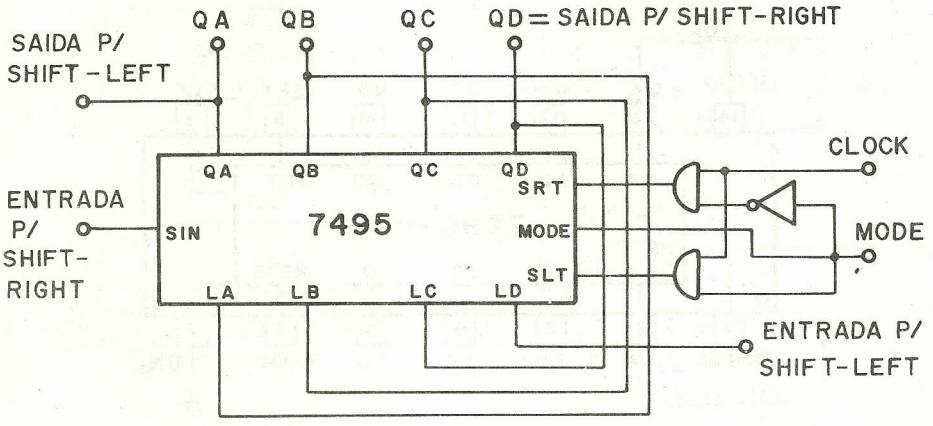


Fig. 8.5 - 7495 Conectado com o SR bidirecional.

A freqüência máxima do sinal de CLOCK para o 7495 é de 36 MHZ.

7496 - 5-BIT SHIFT REGISTER (SHIFT-RIGHT, PARALLEL-IN, PARALLEL-

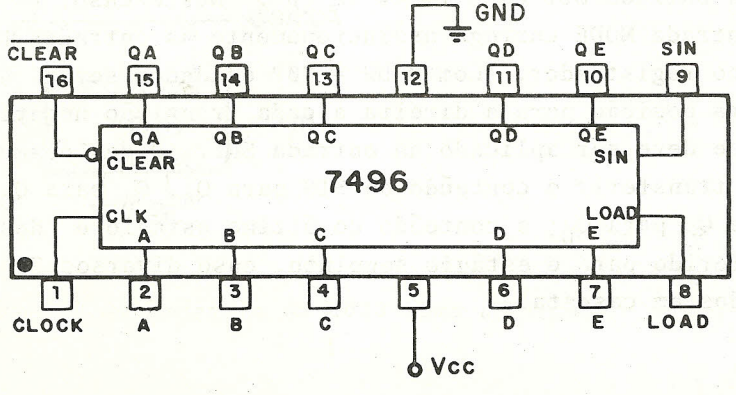


Fig.

Este CI contém um SR de 5 bits com entrada e saída de dados em paralelo. Para operação normal, a entrada $\overline{\text{CLEAR}}$ deve ficar em "1" e a entrada LOAD em "Ø". Se $\overline{\text{CLEAR}}$ vai para "Ø" todos os estágios são ressetados. Se LOAD vai para "1", com $\overline{\text{CLEAR}}$ em "1", os estágios cuja entrada correspondente contiver "1" serão setados e os demais permanecerão inalterados.

Em outras palavras, as entradas paralelas permitem apenas carregar "1"'s; uma operação de CLEAR é necessária antes do LOAD quando se quer carregar um "Ø".

Com $\overline{\text{CLEAR}}$ = "1" e LOAD = "Ø" a informação é deslocada de uma posição para a direita a cada transição positiva do sinal de CLOCK.

A freqüência máxima de operação do 7496 é de 10 MHZ.

74164 - 8-BIT SHIFT REGISTER (SERIAL INPUT, PARALLEL OUTPUT)

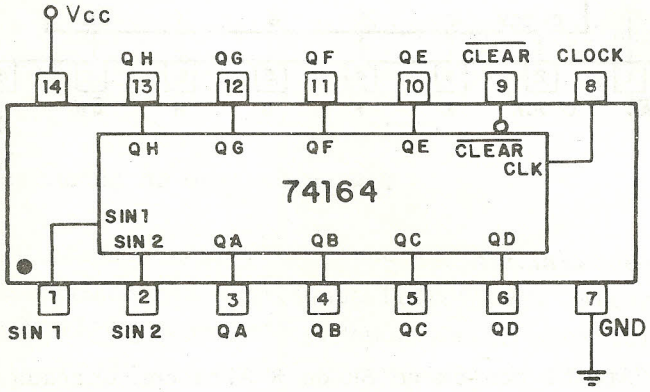


Fig. 8.7 - 74164.

Este CI contém um SR de 8 bits com entrada de dados paralela.

Para operação normal a entrada $\overline{\text{CLEAR}}$ deve ser igual a "1", um nível Ø nesta entrada resseta todos os estágios para "Ø".

As mudanças de estado ocorrem na transição positiva do

sinal de CLOCK. Um nível "Ø" em qualquer das entradas série faz com que um "Ø" seja transferido para o primeiro estágio (Q_A) na transição positiva do CLOCK; um nível "1" em ambas as entradas série faz com que Q_A seja setada na transição positiva do CLOCK. Assim, normalmente uma das entradas série é conectada permanentemente em "1" e a outra é utilizada para entrada de dados.

A freqüência máxima de operação do 74164 é de 36 MHZ.

74165 - 8-BIT SHIFT REGISTER (PARALLEL INPUT, SERIAL OUTPUT)

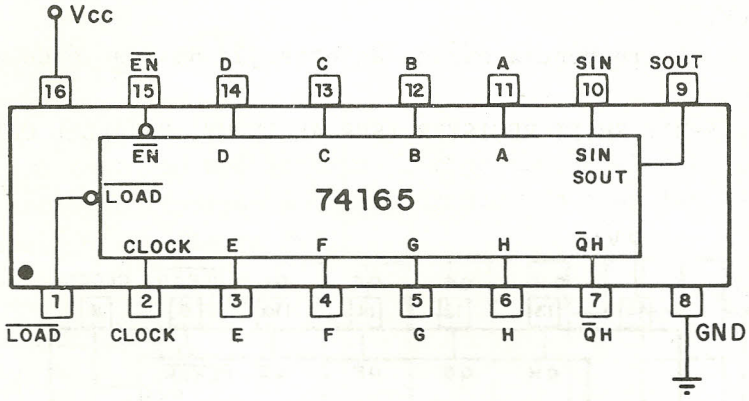


Fig. 8.8 - 74165.

Este CI contém um SR de 8 bits com entrada de dados em paralelo e saída em série.

Para operação normal a linha $\overline{EN}$ deve ficar em "Ø" e $\overline{LOAD}$ em "1". Os dados são deslocados na transição positiva do CLOCK.

Se a entrada $\overline{LOAD}$ vai para "Ø", o conteúdo das entradas A até H é carregado no registrador.

Com $\overline{EN}$ = "Ø" e $\overline{LOAD}$ = "1" os dados são deslocados de uma posição para a direita a cada transição positiva do CLOCK.

A saída complementar do último estágio ($\overline{Q}_H$) está disponível no pino 7.

A freqüência máxima do CLOCK é de 26 MHZ.

8.4 - Registradores de Deslocamento da Familia CMOS

Também na série 4ØØØ CMOS encontramos diversos circuitos integrados MSI que realizam a função de SR's. A seguir a descrição de alguns deles.

4006 - VARIABLE-LENGTH SHIFT REGISTER (SERIAL-IN, SERIAL OUT)

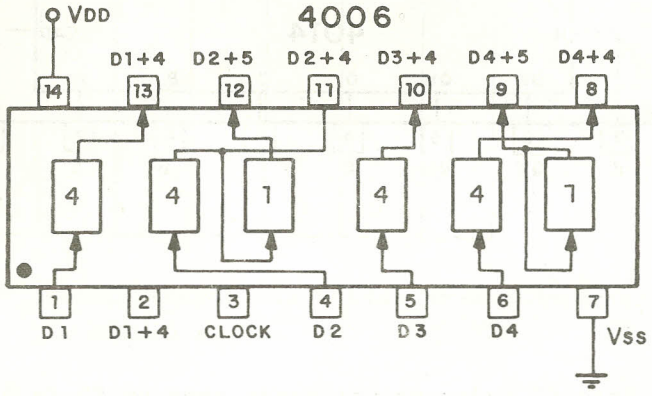


Fig. 8.9 - 4006.

Trata-se de um SR de comprimento variável até um máximo de 18 estágios. O CI contém 4 SR's separados, dois dos quais são de quatro estágios e dois que podem ser usados com quatro ou cinco estágios.

Para todos os SR'S, os dados presentes nas entradas série são deslocados para dentro do registro na transição negativa do sinal de CLOCK.

Os registros podem ser conectados em série permitindo obter comprimentos totais de 4, 5, 8, 9, 10, 12, 13, 14, 16, 17 ou 18 estágios.

Os pinos 8 e 11 não podem ser utilizados como entradas.

A freqüência máxima do CLOCK é de 5 MHZ para V_{cc} =10V e 2,5 MHZ para V_{cc} = 5V.

4014 - 8-BIT SHIFT REGISTER (PARALLEL-IN/SERIAL-OUT, CLOCKED LOAD)

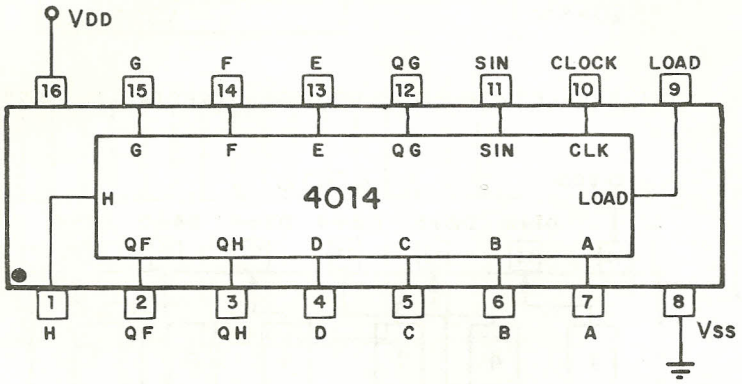


Fig. 8.10 - 4014.

Este CI pode ser utilizado como um SR de 6, 7 ou 8 estágios, tanto como SISO ou PISO.

Quando utilizado no modo SISO, a entrada LOAD deve permanecer em "Ø". A cada transição positiva do CLOCK a informação contida no registro é deslocada de uma posição para a direita. As saídas dos três últimos estágios (Q_F, Q_G e Q_H) estão disponíveis e, portanto, pode-se utilizar o CI como SR de 6,7 ou 8 estágios.

Para carregar as entradas paralelas (A até H) no registrador, a entrada LOAD deve ir para "1". O carregamento é síncrono e ocorre na transição positiva do CLOCK. A entrada LOAD deve retornar para "Ø" e assim permanecer, para que ocorra o deslocamento da informação.

A freqüência máxima do CLOCK é de 5 MHZ para Vcc =10V e 2,5 MHZ para Vcc = 5V.

4015 - DUAL 4-BIT SHIFT REGISTER (SERIAL-IN,PARALLEL OUT)

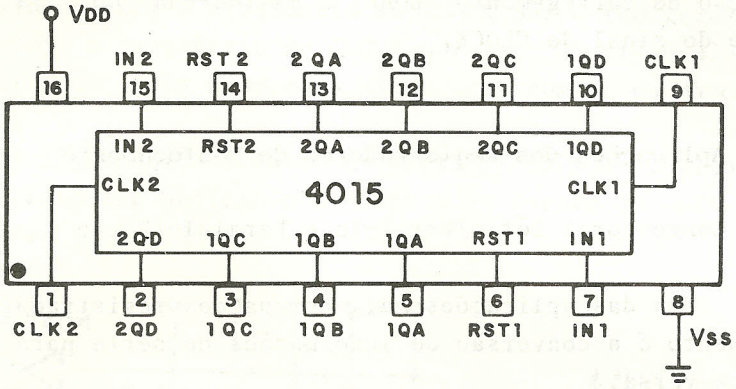


Fig. 8.11 - 4015.

Este CI contém dois SR's de 4 estágios, os dois registros são independentes e possuem entrada de dados série e saída de dados paralela.

Para operação normal, a linha RST deve ficar em "Ø"; um nível lógico "1" nesta linha resseta o registro correspondente para "Ø".

A informação é deslocada na transição positiva do CLOCK.

A freqüência máxima do CLOCK é de 5 MHZ para V_{CC} = 10V e 2,5 MHZ para V_{CC} = 5V.

4021 - 8-BIT SHIFT REGISTER
(PARALLEL-IN, SERIAL OUT, ASSYNCHRONOUS LOAD)

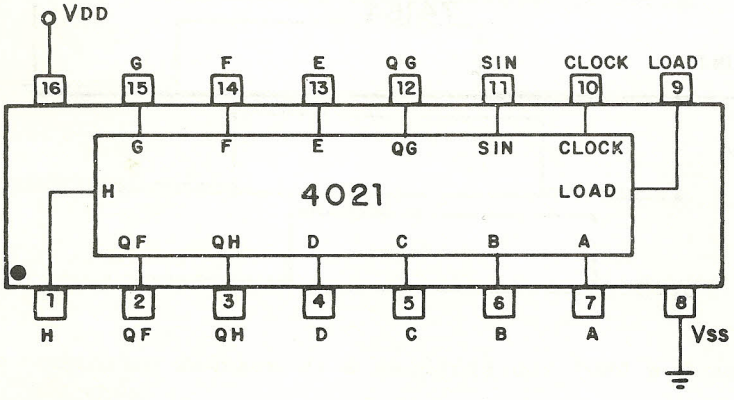


Fig. 8.12 - 4021.

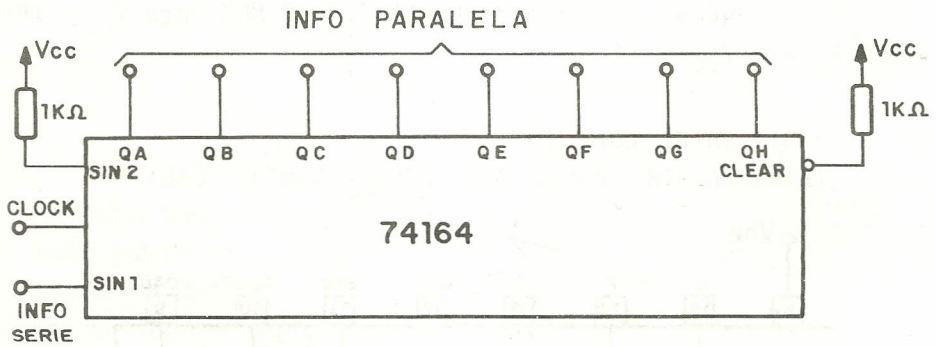
Este CI é similar ao 4014, com a única diferença que a operação de carregamento (LOAD) é assíncrona, ou seja, independente do sinal de CLOCK.

8.5 - Aplicações dos Registradores de Deslocamento

8.5.1 - Conversores Série/Paralelo e Paralelo/Série

Uma das aplicações mais comuns dos registradores de deslocamento é a conversão de informações de série para paralelo e vice-versa.

Vamos supor que uma informação seja transmitida serialmente através de uma linha de dados e que se deseja convertê-la em uma informação paralela, ou seja, na qual todos os bits estejam presentes ao mesmo tempo. Esta tarefa poderia ser feita utilizando um SR, conforme ilustra a figura 8.13, na qual se representa, a conversão de uma seqüência de 8 bits transmitidos seqüencialmente em um byte (palavra de 8 bits) acumulado no registrador:



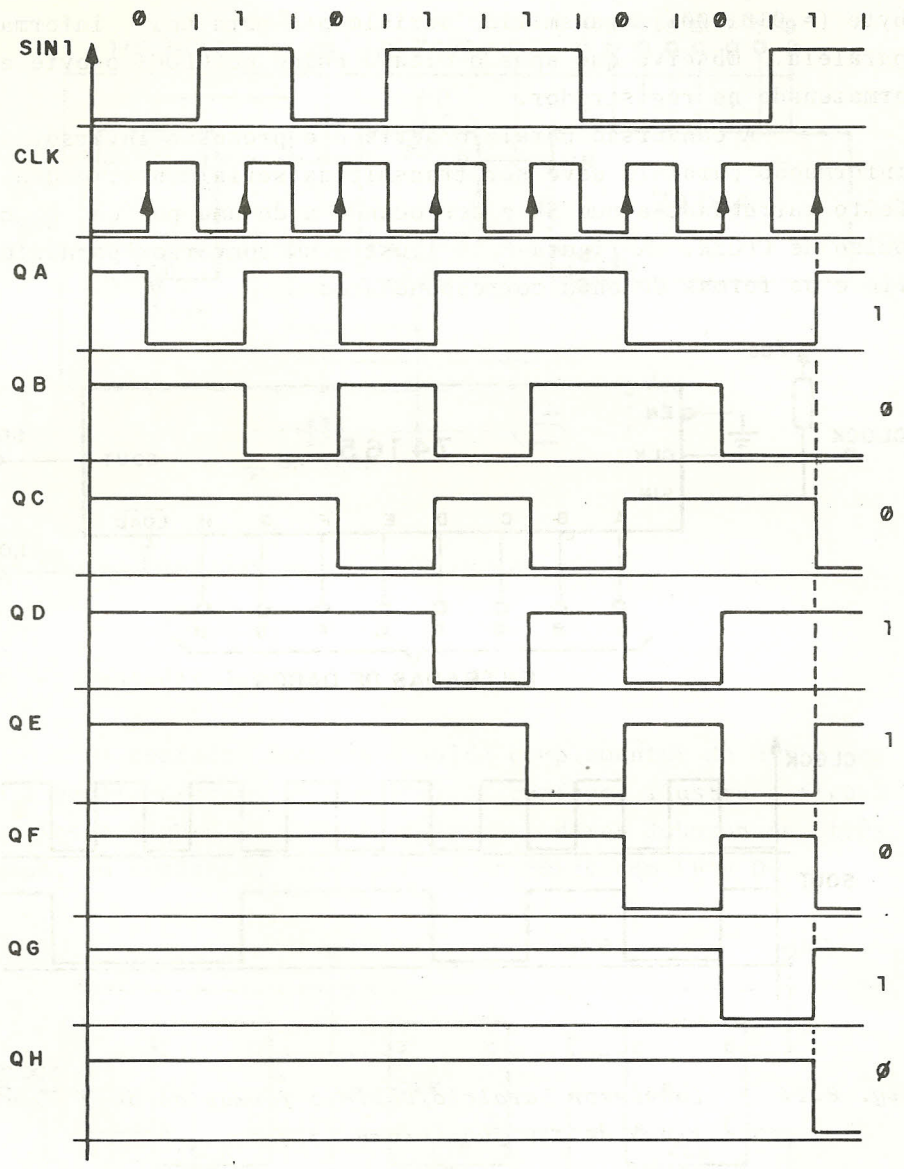


Fig. 8.13 - Conversor Série/Paralelo e formas de onda ilustrativas de seu funcionamento.

Na figura 8.13 está representada a conversão de um

byte (= Ø1Ø11ØØ1) transmitido serialmente para uma informação paralela. Observe que após o oitavo pulso de CLOCK o byte está armazenado no registrador.

A conversão paralelo/série é o processo inverso. A informação paralela deve ser transmitida serialmente, o que é feito carregando-a num SR e deslocando-a de uma posição a cada pulso de CLOCK. A figura 8.14 ilustra um conversor paralelo/série e as formas de onda correspondentes:

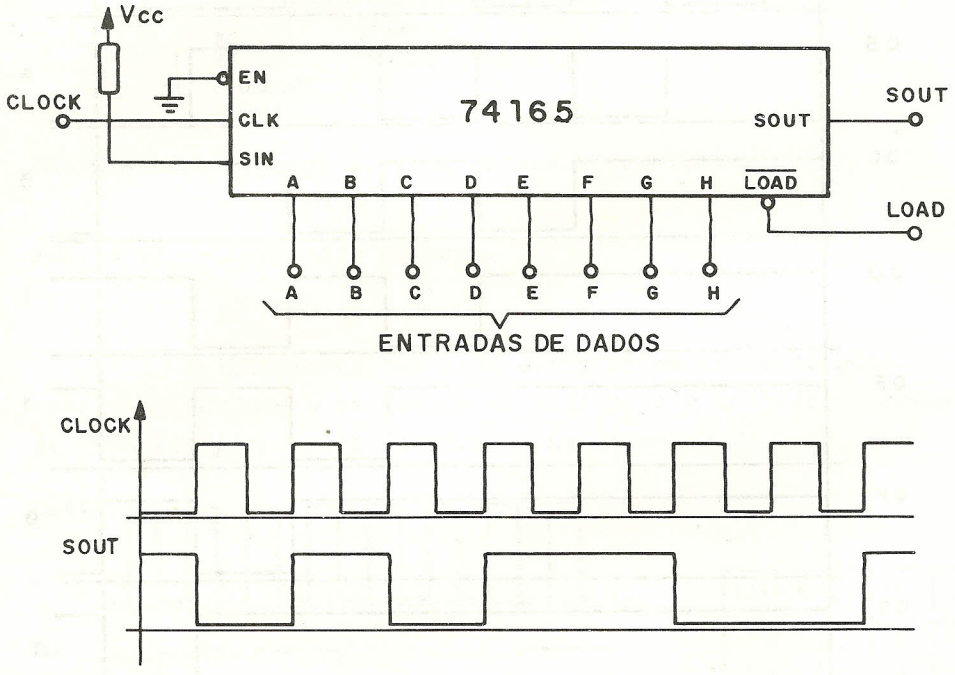


Fig. 8.14 - Conversor Paralelo/Série e formas de onda ilustrativas de seu funcionamento.

Na figura 8.14 admitimos que o byte Ø1Ø11ØØ1 houvesse sido carregado no registrador antes de se iniciar o deslocamento,

A transmissão de informações na forma serial e, portanto, os circuitos de conversão série/paralelo e paralelo/ série são amplamente utilizados em eletrônica digital. A comunicação

entre um microprocessador e um terminal de vídeo, por exemplo, é feita desta maneira. Na figura 8.15, mostra-se um diagrama em blocos para exemplificar a comunicação entre dois dispositivos, através de uma linha serial:

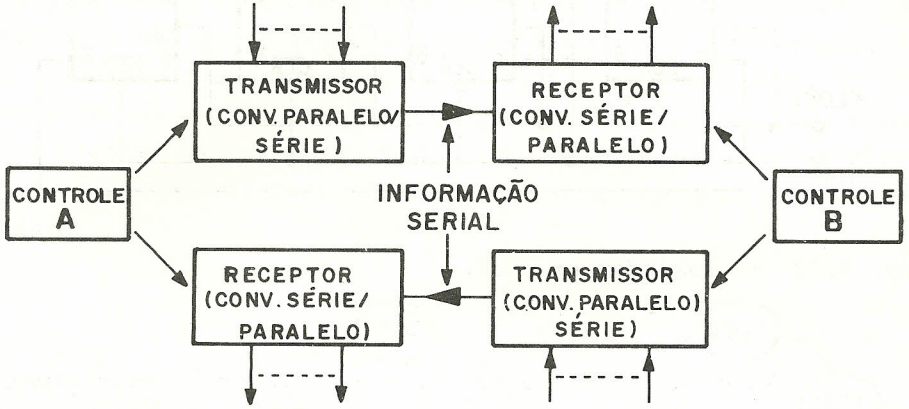


Fig. 8.15 - Comunicação de dados através de linha serial (FULL-DUPLEX).

8.5.2 - Contador em Anel:

Se a saída do último estágio de um SR for realimentada para a entrada do primeiro, obteremos um contador ou seqüenciador.

Uma das possibilidades é ilustrada na figura 8.16 e recebe o nome de contador em anel:

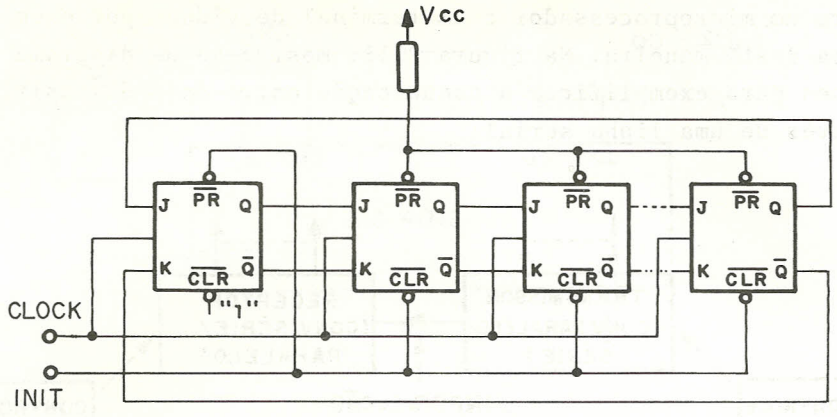


Fig. 8.16 - Contador em anel.

No circuito da figura 8.16 o contador é inicialmente carregado com "1" no primeiro estágio e "Ø" nos demais, através da linha INIT. A cada pulso de CLOCK, o bit "1" será deslocado para a direita e, ao atingir o último estágio, será novamente transferido para o primeiro. Em outras palavras, o contador em anel é aquele no qual um e somente um dos estágios contém "1" (ou "Ø") e todos os demais contém "Ø" (ou "1") a cada instante. O bit "1" (ou "Ø") é circulado pelo contador. A vantagem do contador em anel é que ele fornece a saída já decodificada, uma vez que somente uma das saídas estará em "1" (ou "Ø")para cada contagem. A figura 8.17 mostra a tabela da verdade para um contador em anel de N estágios, onde um bit "1" circula pelo contador:

PULSO DE CLOCK	Q_1	Q_2	Q_3	....	Q_{N-1}	Q_N
0	1	0	0		0	0
1	0	1	0		0	0
2	0	0	1		0	0
⋮						
N-1	0	0	0		0	1
N	1	0	0		0	0

Fig. 8.17 - Tabela da Verdade para o contador em anel.

Conforme vimos anteriormente, é necessário que o contador em anel seja iniciado com um valor correto, o que na figura 8.16 era feito controlando a linha INIT. É interessante que o contador seja construído de maneira a iniciar-se automaticamente quando for aplicada a alimentação (SELF STARTING). Um outro problema é a possibilidade de que o contador entre em uma seqüência não permitida e não saia mais dela. Por exemplo, se um ruído fizer com que um outro estágio fique no estado "1", o contador não mais voltará para a seqüência correta. A figura 8.18 ilustra as possíveis seqüências de contagem para um circuito com quatro estágios. Destas seqüências apenas uma é a desejada.

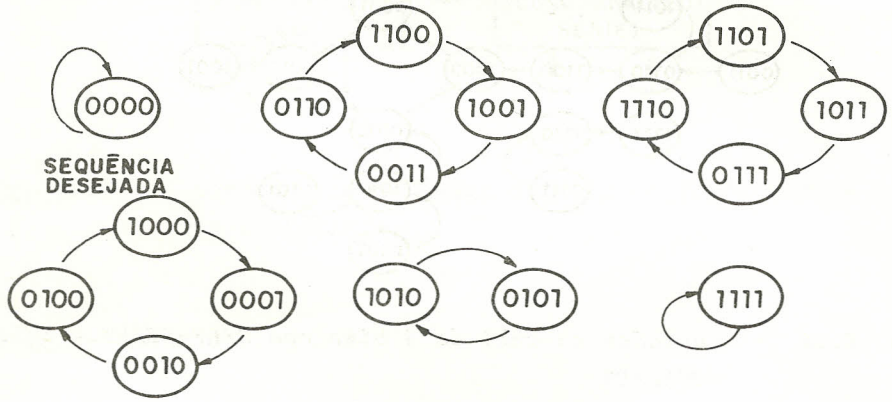


Fig. 8.18 - Seqüências de contagem possíveis para o contador em anel.

É desejável que o circuito consiga retornar por si mesmo à seqüência correta (SELF-CORRECTING). A figura 8.19 ilustra o esquema de um contador em anel de 4 bits capaz de auto-iniciar-se e auto-corrigir-se e as possíveis seqüências de estado para o mesmo:

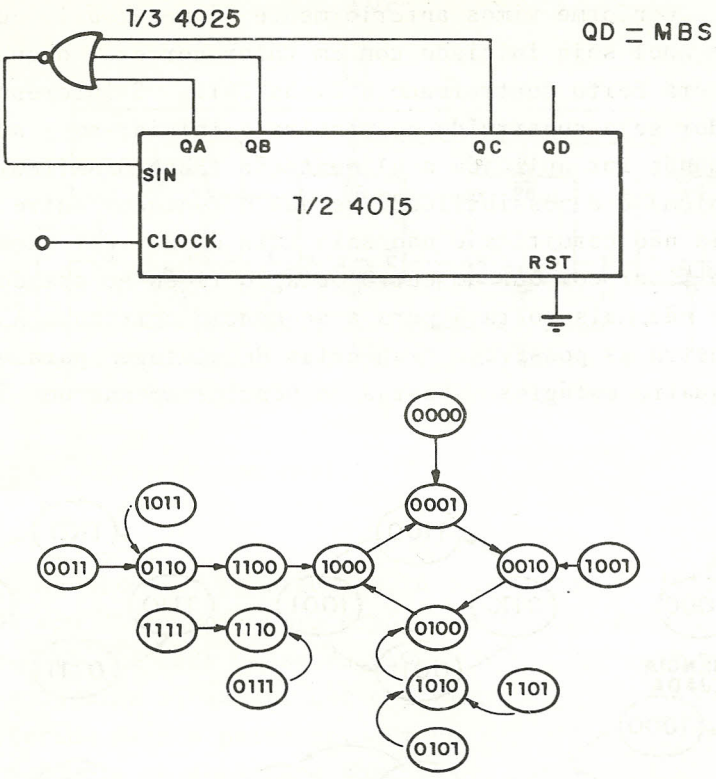


Fig. 8.19 - Contador em anel de 4 bits com SELF-START e SELF--CORRECT.

Pelo diagrama de seqüências apresentado na figura 8.19 verificamos que o contador entrará na seqüência correta em no máximo quatro pulsos de CLOCK.

Na figura 8.20, mostra-se um contador em anel de 10 estágios com o circuito para SELF-START e SELF-CORRECT. Ao ser aplicada a alimentação o primeiro estágio é pressetado para "1" e os demais ressetados para "Ø", sempre que um bit "1" for transferido para o primeiro FF os demais estágios são ressetados, o que faz qualquer seqüência incorreta convergir para a seqüência desejada.

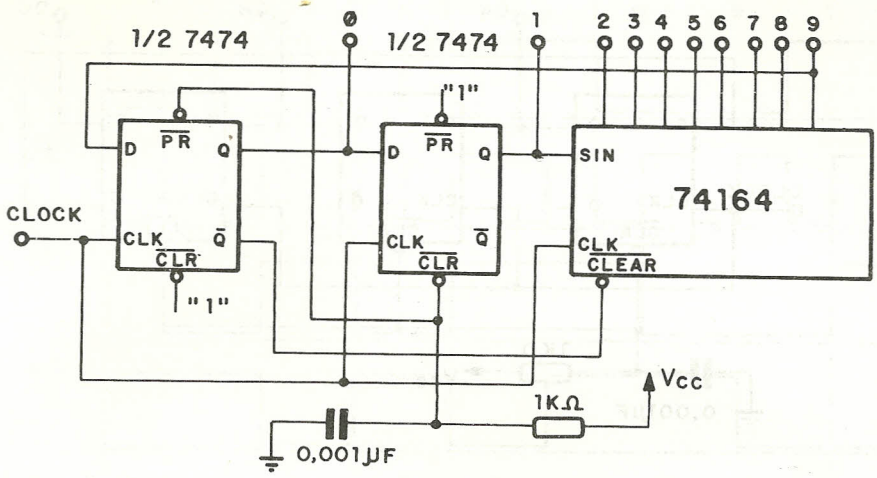
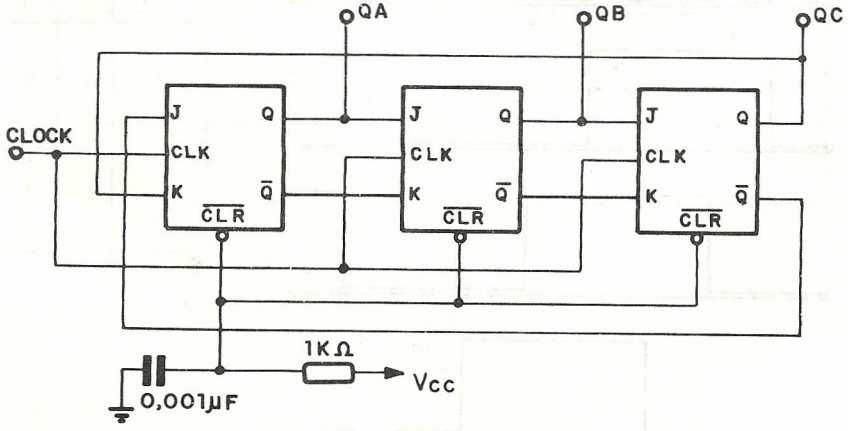


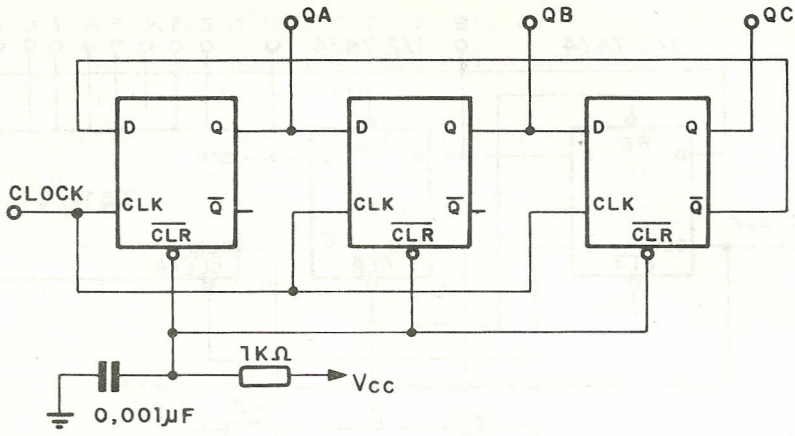
Fig. 8.20 - Contador em anel de 10 estágios.

8.5.3 - Contador Johnson

No contador Johnson a saída complementar do último estágio é realimentada para o primeiro, conforme ilustra a figura 8.21. Nesta figura mostram-se dois contadores Johnson de três estágios, um construído com FF's J-K e outro com FF's D.



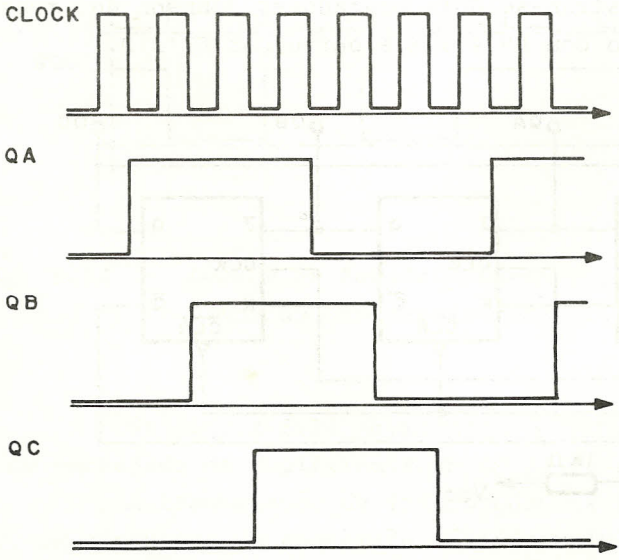
(a) Contador Johnson de 3 estágios com FLIP-FLOPS J-K.



(b) Contador Johnson de três estágios com FLIP-FLOPS D.

Fig. 8.21 - Contador Johnson.

Inicialmente, os três estágios estão ressetados. A tabela da verdade e as formas de onda para o contador Johnson de três estágios estão ilustrados na figura 8.22.



CLOCK	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	1
4	1	1	0
5	1	0	0

Fig. 8.22 - Formas de Onda e tabela da verdade para o contador Johnson de três estágios (módulo 6).

Conforme se vê pelas formas de onda, o contador Johnson de 3 estágios divide a freqüência do CLOCK por 6 (em geral, n estágios dividem a freqüência do CLOCK por 2n); as três saídas Q_A, Q_B e Q_C fornecem ondas quadradas simétricas com $1/6$ da freqüência do CLOCK.

A figura 8.23 ilustra um contador Johnson divisor por 10 (5 estágios) que utiliza o CI 7496 e as formas de onda correspondentes:

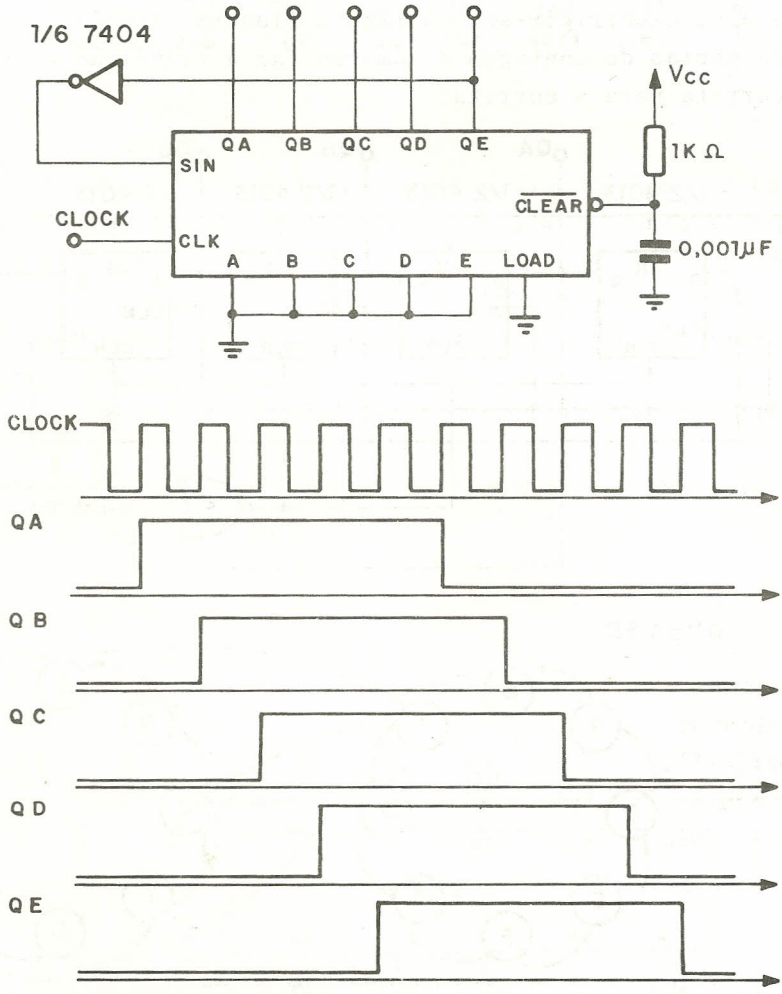
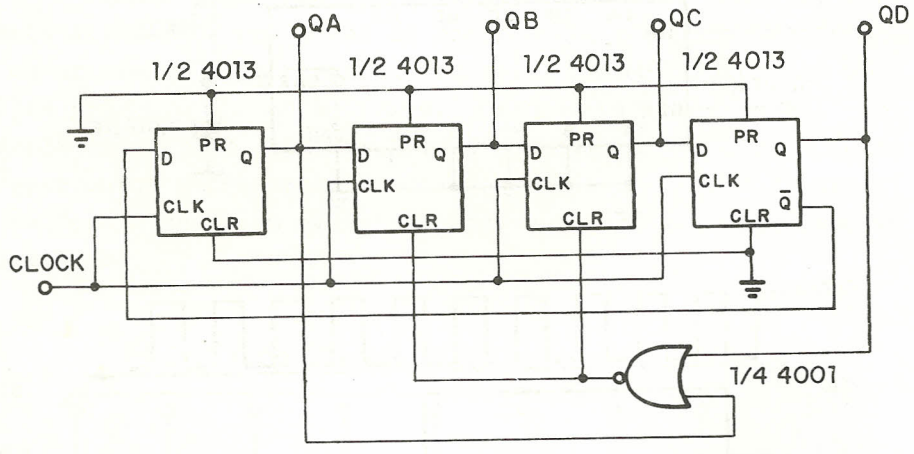
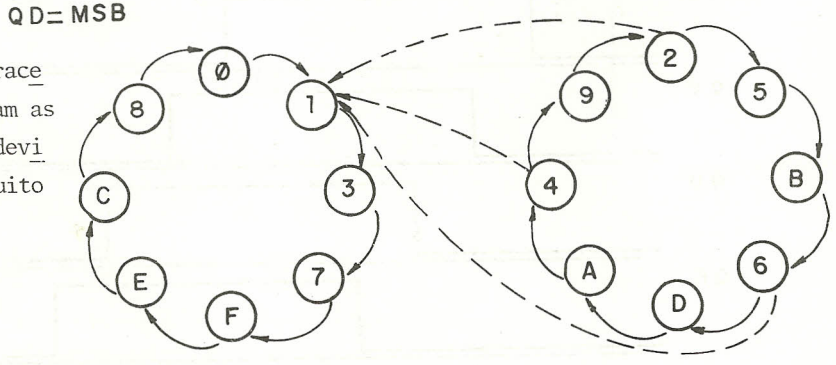


Fig. 8.23 - Contador Johnson de 5 estágios com 7496 e formas de onda.

Um contador Johnson de n estágios terá 2^n-2n estados não permitidos. Para contornar o problema dos estados não permitidos, evitando que o contador entre e fique preso em uma seqüência incorreta pode-se utilizar as saídas do primeiro e do último estágios para gerar um sinal de CLEAR dos demais FLIP-FLOPS, quando este primeiro e último estágios forem iguais a "Ø".

A figura 8.24 ilustra um contador Johnson de 4 estágios que utiliza este esquema; o contador é capaz de auto iniciar-se e auto-corrigir-se. A figura ilustra, também, as possíveis seqüências de contagem e como se faz a transição da seqüência incorreta para a correta:





as linhas tracejadas indicam as transições devidas do circuito de CLEAR.

Fig. 8.24 - Contador Johnson de 4 estágios com SELF-START e SELF-CORRECT.

Uma vantagem do contador Johnson é que a decodificação de seus estados pode ser feita sempre com portas de duas entradas apenas. Considere por exemplo um contador Johnson de módulo 6 (3 estágios) e sejam S_ϕ, S_1, S_2, S_3, S_4 e S_5 as saídas decodificadas. Apenas uma delas estará ativa para cada estado do contador. A figura 8.25 ilustra a tabela da verdade para o contador e as saídas decodificadas, bem como as equações lógicas resultantes:

Q_C	Q_B	Q_A	S_ϕ	S_1	S_2	S_3	S_4	S_5
0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0	0
1	1	1	0	0	0	1	0	0
1	1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1

$$S_\phi = \overline{Q}_C \cdot \overline{Q}_A$$
$$S_1 = \overline{Q}_B \cdot Q_A$$
$$S_2 = \overline{Q}_C \cdot Q_B$$
$$S_3 = Q_C \cdot Q_A$$
$$S_4 = Q_B \cdot \overline{Q}_A$$
$$S_5 = Q_C \cdot \overline{Q}_B$$

Fig. 8.25 - Decodificação das saídas para o contador Johnson de módulo 6.

O processo ilustrado na figura 8.25 pode ser aplicado para contadores de módulos maiores e sempre se conseguirá decodificar as saídas com portas lógicas de duas entradas apenas.

O contador Johnson pode ser modificado para alcançar um ciclo de contagem com um número ímpar de estados. A figura 8.26 ilustra um contador Johnson de módulo 3, as formas de onda correspondentes e as equações das saídas decodificadas:

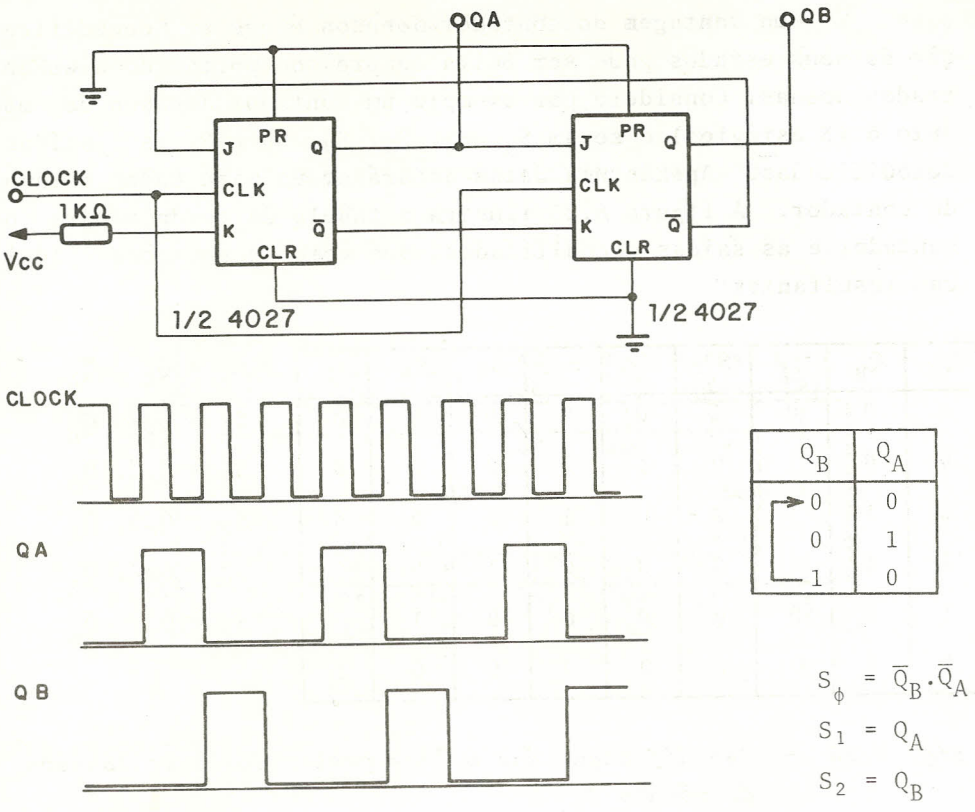
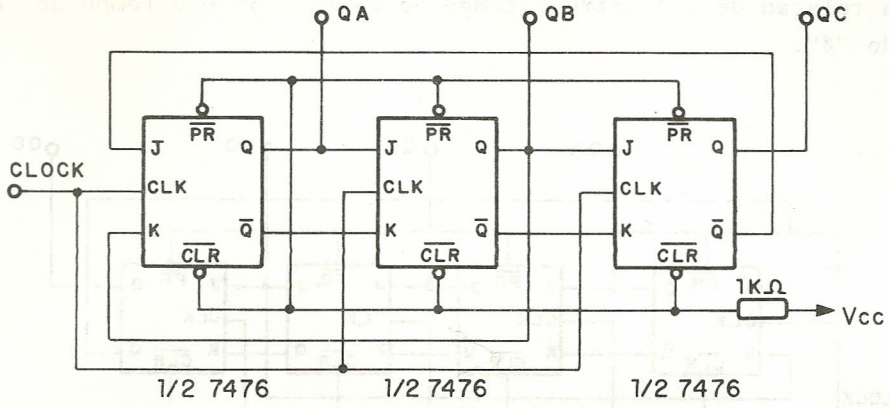
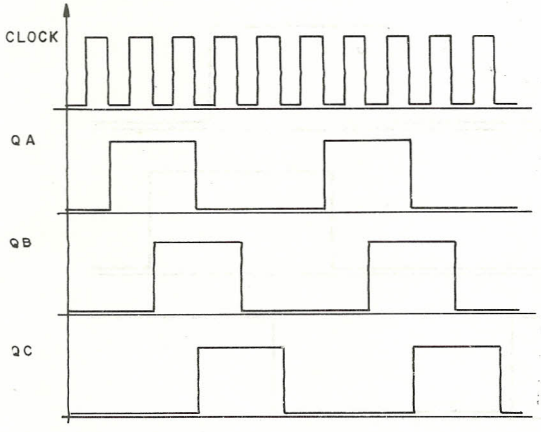


Fig. 8.26 - Contador Johnson de módulo-3, formas de onda, tabela de estados e equações das saídas decodificadas.

Para contadores de módulo ímpar até 13, não há necessidade de nos preocuparmos com os estados não permitidos, pois a seqüência correta será sempre atingida, independentemente do ponto de partida.

As figuras 8.27 e 8.28 ilustram contadores de módulos 5 e 7, respectivamente. Num contador de módulo ímpar o primeiro estágio pelo menos deve ser um FLIP-FLOP J-K.





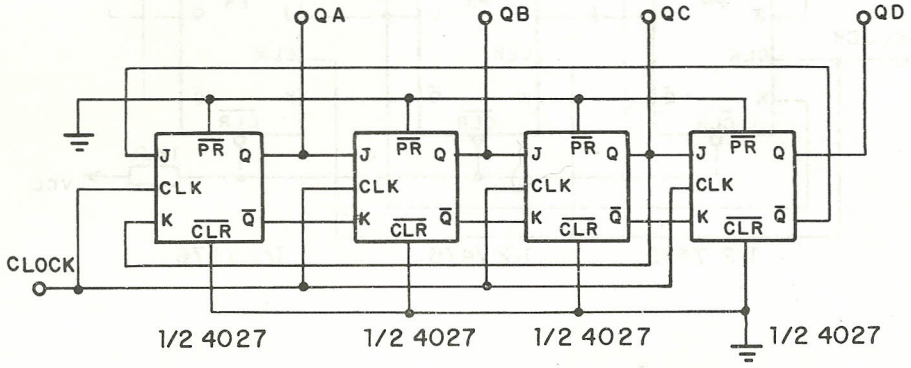
Q_C	Q_B	Q_A
0	0	0
0	0	1
0	1	1
1	1	0
1	0	0

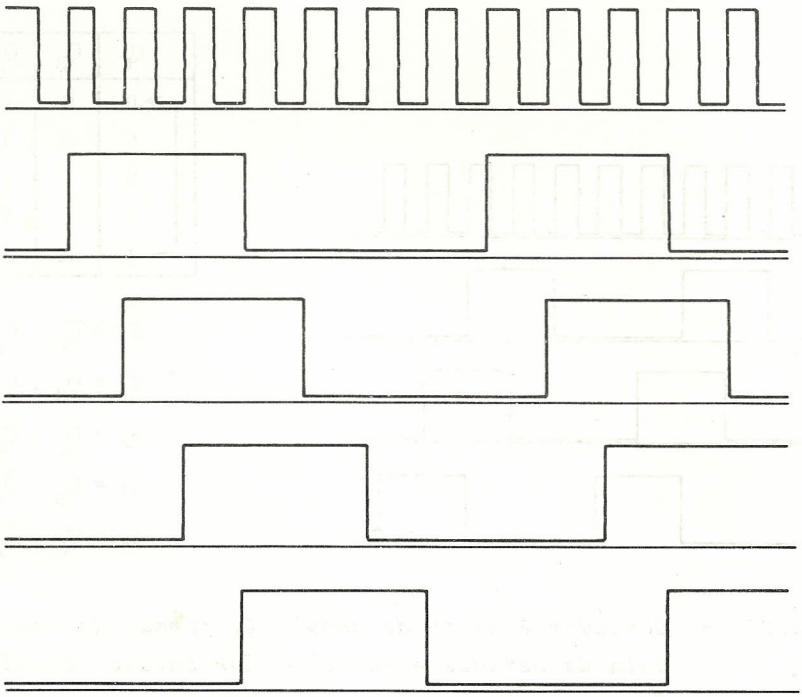
$$S_\phi = \overline{Q}_C \cdot \overline{Q}_A$$
$$S_1 = \overline{Q}_B \cdot Q_A$$
$$S_2 = \overline{Q}_C \cdot Q_B$$
$$S_3 = Q_B \cdot \overline{Q}_A$$
$$S_4 = Q_C \cdot \overline{Q}_B$$

Fig. 8.27 - Contador Johnson de módulo 5, formas de ondas, tabela de estados e equações das saídas decodificadas.

Observe que no contador da figura 8.27 as saídas têm

uma relação de 2:3 entre o tempo no estado "1" e o tempo no estado "Ø".





Q_D	Q_C	Q_B	Q_A
0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0

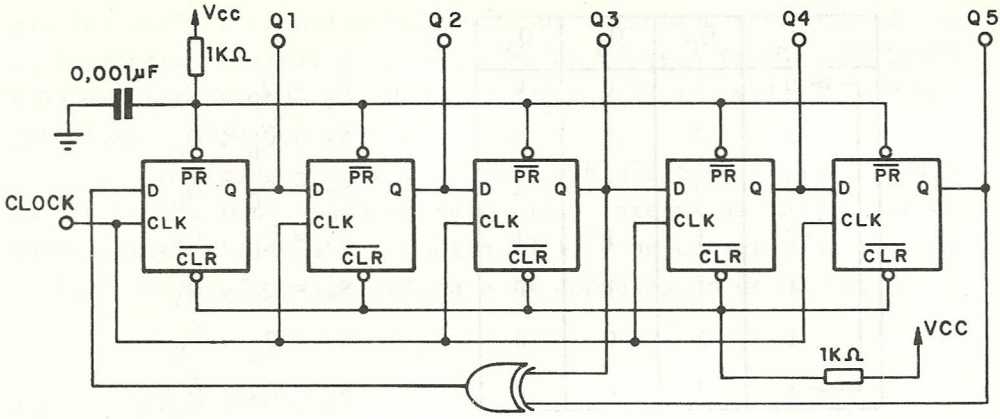
$$S_\phi = \overline{Q}_D \cdot \overline{Q}_A$$
$$S_1 = \overline{Q}_B \cdot Q_A$$
$$S_2 = \overline{Q}_C \cdot Q_B$$
$$S_3 = \overline{Q}_D \cdot Q_C$$
$$S_4 = Q_B \cdot \overline{Q}_A$$
$$S_5 = Q_C \cdot \overline{Q}_B$$
$$S_6 = Q_D \cdot \overline{Q}_C$$

Fig. 8.28 - Contador Johnson de módulo 7, formas de onda, tabela de estados e equações das saídas decodificadas.

8.5.4 - Gerador de Seqüência Pseudoaleatória

Utilizando a realimentação da saída de alguns estágios para a entrada série de um SR é possível obter uma seqüência de certo número de estados. Para que esta seqüência seja eficiente, é interessante que seu comprimento seja o maior possível. Um circuito com n estágios possui 2^n estados possíveis e, num SR com realimentação, o comprimento máximo da seqüência será 2^n-1, os circuitos que fornecem uma seqüência de estados com este comprimento máximo são denominados do tipo MLS (MAXIMUM LENGTH SEQUENCE).

A figura 8.29 ilustra um SR de 5 estágios com a realimentação obtida através do OU-ESCLUSIVO das saídas Q_3 e Q_5 e a tabela de estados correspondente:



	Q_5	Q_4	Q_3	Q_2	Q_1		Q_5	Q_4	Q_3	Q_2	Q_1	
(1)	0	0	0	0	1	→	1	1	0	0	0	(24)
(2)	0	0	0	1	0		1	0	0	0	1	(17)
(4)	0	0	1	0	0		0	0	0	1	1	(3)
(9)	0	1	0	0	1		0	0	1	1	0	(6)
(18)	1	0	0	1	0		0	1	1	0	1	(13)
(5)	0	0	1	0	1		1	1	0	1	1	(27)
(11)	0	1	0	1	1		1	0	1	1	1	(23)
(22)	1	0	1	1	0		0	1	1	1	0	(14)
(12)	0	1	1	0	0		1	1	1	0	1	(29)
(25)	1	1	0	0	1		1	1	0	1	0	(26)
(19)	1	0	0	1	1		1	0	1	0	1	(21)
(7)	0	0	1	1	1		0	1	0	1	0	(10)
(15)	0	1	1	1	1		1	0	1	0	0	(20)
(31)	1	1	1	1	1		0	1	0	0	0	(8)
(30)	1	1	1	1	0		1	0	0	0	0	(16)
(28)	1	1	1	0	0							

Fig. 8.29 - Gerador de seqüência Pseudo-aleatória de 5 estágios e tabela de estados.

Na tabela da figura 8.29, os números entre parênteses representam o equivalente decimal do conteúdo do registrador. Observa-se que os números estão bem misturados e sem uma ordem definida. Assim, diz-se que o circuito gera uma seqüência,pseu

do-aleatória, pois embora ela se repita após 2^n-1 CLOCKS, onde n é o número de estágios, parecerá aleatória se consideramos apenas um grupo de estados. A tabela da figura 8.30 mostra como se deve ligar a realimentação de um gerador de seqüência pseudo-aleatória para que se obtenha uma seqüência de máximo comprimento.

n	COMPRIMENTO DA SEQUÊNCIA	USAR SAÍDA DOS ESTÁGIOS
2	3	1,2
3	7	2,3
4	15	3,4
5	31	3,5
6	63	5,6
7	127	6,7
8	255	4,5,6,8
9	511	5,9
10	1023	7,10
11	2047	9,11
12	4095	6,8,11,12
13	8191	9,11,12,13
14	16383	4,8,13,14
15	32767	14,15
16	65535	4,13,15,16
17	131071	14,17
18	262143	11,18
19	524387	14,17,18,19
20	1048575	17,20
21	2097151	19,21
22	4194303	21,22
23	8388607	18,23
24	16777215	17,22,23,24
25	33554431	22,25
26	67108363	20,24,25,26
27	134217727	22,25,26,27
28	268435455	25,28
29	536870911	27,29
30	1073741823	7,28,29,30
31	2147483647	28,31

Fig. 8.30 - Tabela de conexões para gerador de seqüência pseudo-aleatória.

Para um SR de n estágios podemos obter, no mínimo, quatro seqüências do tipo MLS:

a) direta normal - a realimentação é feita com o OU-EXCLUSIVO das saídas indicadas na tabela da figura 8.30, é o caso ilustrado na figura 8.29;

b) direta complementar - a realimentação é feita com o OU-EXCLUSIVO invertido (operação de equivalência) das saídas, indicadas na tabela;

c) reversa normal - a realimentação é feita com o OU-EXCLUSIVO das saídas (n-número indicado na tabela); se o resultado é "Ø" toma-se a saída n. Por exemplo, para 6 estágios (n=6) usam-se as saídas 1 (6-5) e 6 (6-6);

d) reversa complementar - a realimentação é feita como no caso c, mas usando a operação OU-EXCLUSIVO invertida.

Como a seqüência de máximo comprimento possui 2^n-1 estados, há um estado não permitido. Para as seqüências normais é o estado ØØ...Ø e para as seqüências complementares é o estado 11...1. Observe que, de fato, se todos os estágios do circuito da figura 8.29 estiverem ressetados, o registrador não sairá deste estado. Assim, é conveniente iniciar o gerador para um estado permitido, o que é feito no circuito da figura 8.29 pressetando todos os FF's quando a alimentação é aplicada.

A figura 8.31 ilustra um gerador de seqüência pseudo-aleatória de 16 bits, gerando uma seqüência reversa complementar de máximo comprimento, com deteção e correção do estado não-permitido:

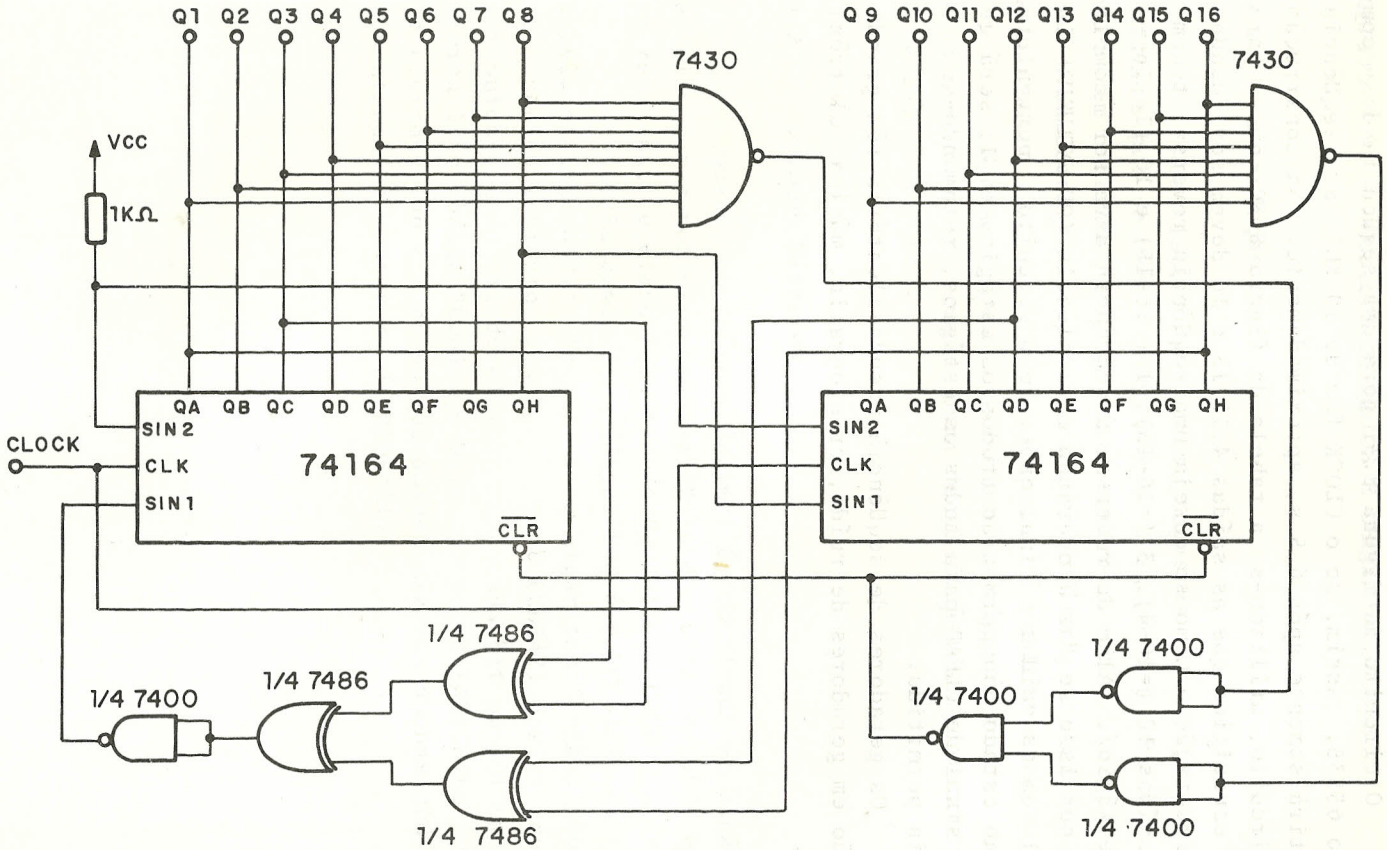


Fig. 8.31 - Gerador de seqüência pseudo-aleatória de 16 bits com circuito de auto-correção.

O circuito da figura 8.31 gera uma seqüência de comprimento 65535. Assim, se o CLOCK for de 10 KHZ, a seqüência se repetirá somente após 6,5 s, aproximadamente. Na construção deste circuito, utilizou-se a tabela da figura 8.30, que para 16 bits especifica que as saídas 4,13,15 e 16 devem ser usadas na realimentação. Como se deseja uma seqüência reversa, tomam-se as saídas 12 (=16-4), 3 (=16-13), 1 (16-15) e 16 (16-16==> último estágio). Além de reversa, a seqüência deve ser complementar, por isso se faz a operação equivalência (OU-EXCLUSIVO invertida) com as saídas. Finalmente, se o circuito porventura entrar no estado não permitido (todos os estágios em 1) será gerado um sinal de CLEAR para todos os estágios, retomando-se a seqüência permitida.

Os geradores de seqüência pseudo-aleatória têm grande aplicação em geradores de ruído, criptografia, música eletrônica, etc.

CAPÍTULO 9 - CÓDIGOS BINÁRIOS E DECODIFICADORES

9.1 - Introdução

Nos sistemas digitais, todas as informações são representadas por seqüências de "Ø"s e "1"s, vale dizer, são codificadas em binário. Muitas vezes ocorre a necessidade de se traduzir ou decodificar a informação codificada em binário gerando algum outro tipo de informação que será utilizada no circuito.

Neste capítulo, examinaremos alguns códigos binários e os circuitos utilizados para decodificá-los, juntamente com diversas aplicações.

9.2 - Códigos Binários

Quando um número decimal é codificado em binário, ele é feito corresponder a uma seqüência de "Ø"'s e "1"'s e o código resultante é classificado como BCD(BINARY-CODED DECIMAL).

O mais comum dos códigos BCD é o BCD8421, mostrado na tabela da figura 9.1, no qual cada algarismo decimal corresponde a seu equivalente binário. O nome BCD8421 deriva do peso atribuído a cada bit de código; como cada bit possui um valor posicional, o código BCD8421 é dito com pesos (WEIGHTED CODE). Este código é tão difundido que quando dizemos apenas código BCD subentende-se tratar-se do BCD8421.

DECIMAL	BCD8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Fig. 9.1 - Código BCD8421.

A vantagem do código BCD8421 é a facilidade da conversão de decimal para binário e vice-versa. Uma desvantagem, contudo, é que as regras da adição binária não são diretamente aplicáveis a números codificados em BCD8421.

Um outro código BCD relativamente comum é o chamado código excesso-3, ilustrado na tabela da figura 9.2. A vantagem deste código é que ele é auto-complementar (SELF-COMPLEMENTING) significando que um número decimal e seu complemento 9 (o que falta para 9) correspondem a binários também complementares.

Esta propriedade é útil na realização de operações aritméticas com números decimais codificados em BCD excesso-3.

DECIMAL	BCD excesso-3
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Fig. 9.2 - Código BCD excesso-3.

Além dos dois códigos já apresentados, há diversos outros códigos BCD de 4 bits, alguns dos quais são mostrados na figura 9.3, e que possuem aplicação mais restrita.

DECIMAL	7421	6311	5421	5311	5211	4221	3321	2421
0	0000	0000	0000	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0001	0001	0001	0001
2	0010	0011	0010	0011	0011	0010	0010	0010
3	0011	0100	0011	0100	0101	0011	0011	0011
4	0100	0101	0100	0101	0111	1000	0101	0100
5	0101	0111	1000	1000	1000	0111	1010	1011
6	0110	1000	1001	1001	1001	1100	1100	1100
7	1000	1001	1010	1011	1011	1101	1101	1101
8	1001	1011	1011	1100	1101	1110	1110	1110
9	1010	1100	1100	1101	1111	1111	1111	1111

Fig. 9.3 - Alguns Códigos BCD de 4 bits.

Embora 4 bits sejam suficientes para representar todos os algarismos decimais, existem códigos BCD de 5 bits ou mais. Os bits adicionais facilitam a decodificação em alguns casos e permitem a deteção de erros com mais facilidade. O código 2-em-5, por exemplo, faz cada número decimal corresponder a um binário de 5 bits, dos quais exatamente 2 são iguais a 1. Isto torna a paridade (número de 1's na palavra) sempre par e facilita a deteção de erros. O código 2-em-5 é bastante utilizado em telefonia.

Outro código BCD de 5 bits importante é o código Johnson. Conforme vimos no capítulo 8, este código é facilmente decodificável. A mesma característica é apresentada pelo código BCD 51111.

A tabela apresentada na figura 9.4 ilustra alguns códigos BCD de 5 bits:

DECIMAL	2-em-5	63210	Johnson	86421	51111
Ø	00011	00110	00000	00000	00000
1	00101	00011	00001	00001	00001
2	00110	00101	00011	00010	00011
3	01001	01001	00111	00011	00111
4	01010	01010	01111	00100	01111
5	01100	01100	11111	00101	10000
6	10001	10001	11110	01000	11000
7	10010	10010	11100	01001	11100
8	10100	10100	11000	10000	11110
9	11000	11000	10000	10001	11111

Fig. 9.4 - Alguns códigos BCD de 5 bits.

Dos códigos BCD de mais de 5 bits os mais importantes são o biquinário ou 50 43210, o 543210 e o 9876543210 ou código anel. Os dois primeiros permitem fácil deteção de erros; o código anel é o de maior facilidade de decodificação, uma vez que a contagem já é feita decodificada. Estes códigos estão ilustrados na tabela da figura 9.5:

DECIMAL	50 43210	543210	9876543210
0	01 00001	000001	0000000001
1	01 00010	000010	0000000010
2	01 00100	000100	0000000100
3	01 01000	001000	0000001000
4	01 10000	010000	0000010000
5	10 00001	100001	0000100000
6	10 00010	100010	0001000000
7	10 00100	100100	0010000000
8	10 01000	101000	0100000000
9	10 10000	110000	1000000000

Fig. 9.5 - Alguns códigos BCD de mais de 5 bits.

Um código bastante utilizado é o código Gray, ilustrado na tabela da figura 9.6. Este não é um código BCD e sua característica notável é que entre dois números sucessivos apenas um bit varia. Esta particularidade é útil em certas aplicações, como conversores analógico-digitais, controle de servomecanismos, etc. Observe que nos mapas de Karnaugh as variáveis são ordenadas segundo o código Gray, justamente para que entre duas posições adjacentes somente uma das variáveis mude de estado.

DECIMAL	BINÁRIO	GRAY
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000
⋮		

Fig. 9.6 - Código Gray.

Os códigos apresentados até o momento são utilizados para codificar números. Há códigos chamados alfanuméricos que permitem representar caracteres alfabéticos maiúsculos e minúsculos, algarismos, caracteres especiais (?,!,%, etc) e caracteres de controle (CR-CARRIAGE RETURN, LF-LINE FEED, para controle de um teletipo ou terminal de vídeo). Dois destes códigos são

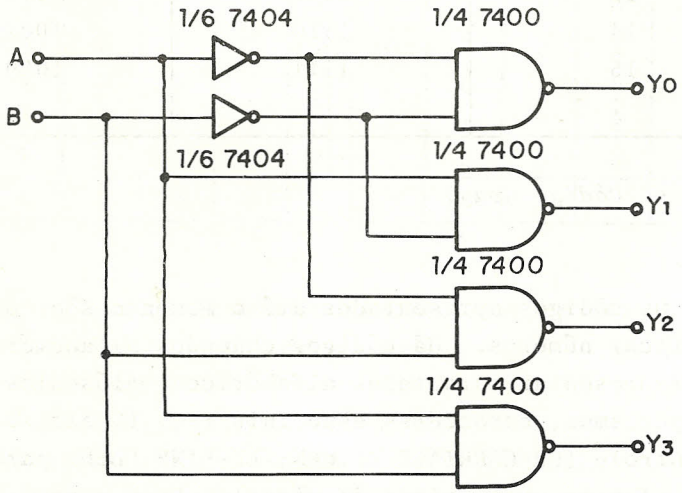
padrões utilizados universalmente nos computadores digitais e equipamentos periféricos. São eles os códigos EBCDIC (EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE) e ASCII (AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE).

O código EDCDIC é um código binário de 8 bits, utilizado principalmente para representação interna de caracteres em computadores digitais e o código ASCII é um código binário de 7 bits, utilizado principalmente em comunicação de dados.

9.3 - Decodificadores

Um decodificador é um circuito que recebe uma informação codificada de alguma forma e a traduz para outra. A informação pode ser um número decimal codificado em binário, um endereço de uma posição de memória, etc.

Uma classe importante de decodificadores são os decodificadores de n para 2^n linhas (2 para 4, 3 para 8, 4 para 16, etc). Estes decodificadores possuem n linhas de entrada e 2^n linhas de saída, das quais uma e somente uma ficará ativa para cada combinação das linhas de entrada. A figura 9.7 mostra um decodificador de 2 linhas para 4 linhas e sua tabela da verdade:



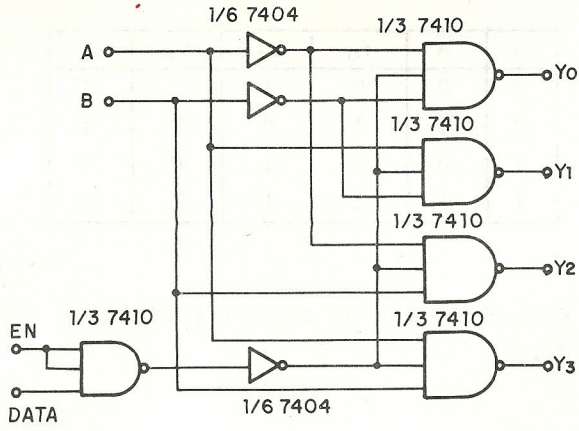
B	A	Y_0	Y_1	Y_2	Y_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Fig. 9.7 - Decodificador de 2 para 4 linhas e tabela da verdade.

Neste circuito vemos que somente a saída correspondente à combinação das entradas A e B estará ativa(neste caso,ativa no estado "Ø") em cada instante.

O conceito apresentado na figura 9.7 pode ser facilmente estendido para 3, 4 ou n linhas de entrada. Na pratica, não é necessário construir estes decodificadores a partir de portas lógicas, pois há uma grande variedade de CI's que realizam estas funções.

O circuito da figura 9.7 pode ser utilizado com algumas modificações para executar uma função muito útil, qual seja, direcionar uma informação a uma dentre 4 linhas de saída conforme o controle nas linhas A e B. A figura 9.8 ilustra o circuito resultante, que se denomina demultiplexador, demultiplexer, data distributor, ou simplesmente DEMUX, de 1 linha para 4 linhas.

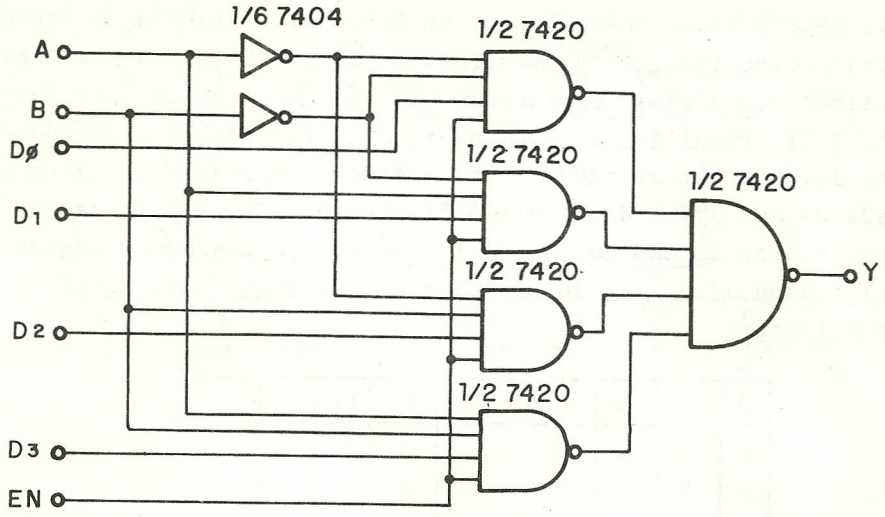


EN	B A D	Y_0	Y_1	Y_2	Y_3
0	X X X	1	1	1	1
1	0 0 0	1	1	1	1
1	0 1 0	1	1	1	1
1	1 0 0	1	1	1	1
1	1 1 0	1	1	1	1
1	0 0 1	0	1	1	1
1	0 1 1	1	0	1	1
1	1 0 1	1	1	0	1
1	1 1 1	1	1	1	0

Fig. 9.8 - DEMUX de 1 linha para 4 linhas e tabela da verdade.

Neste circuito acrescentamos uma linha de habilitação ou ENABLE e podemos verificar que se EN = "Ø" todas as saídas ficam inativas, independente das linhas de controle e da linha de dados (DATA). Se EN = "1", o conteúdo da linha de dados é transferido invertido para a saída endereçada pelas linhas A e B. Há diversos tipos de DEMUX disponíveis na forma de CI's e este circuito tem grande aplicação prática.

É de grande utilidade um circuito que execute a função contrária ao da figura 9.8, ou seja, selecione uma entre 2^n entradas e a direcione para uma linha de saída única. A figura 9.9 ilustra um circuito que realiza esta função e sua tabela da verdade. Ele se denomina MULTIPLEX, DATA SELECTOR ou simplesmente MUX, de quatro linhas para uma linha.



EN	B	A	Y
0	X	X	0
1	0	0	D_ϕ
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3

Fig. 9.9 - MUX de quatro linhas para uma linha e tabela da verdade.

Os três tipos de circuitos apresentados nas figuras 9.7, 9.8 e 9.9 (decodificador de n para 2^n, demultiplex de 1 para 2^n e multiplex de 2^n para 1)são amplamente utilizados em sistemas digitais e encontram-se disponíveis sob forma de CI's nas famílias TTL e CMOS, conforme veremos adiante.

Um decodificador é essencialmente um circuito combinacional e, assim sendo, utilizamos no projeto as técnicas apresentadas no capítulo 4. Para exemplificar, tomemos um caso pouco usual e projetemos um decodificador de BCD8421 para BCD5211. Teremos neste caso um circuito com quatro entradas D, C, B e A representando o número em BCD8421 e quatro saídas Y_3, Y_2, Y_1 e

Y_0, representando o resultado em BCD5211. A tabela da verdade para o circuito, está mostrada na figura 9.10. Nesta tabela, fizemos com que as combinações não válidas para as entradas(1010 até 1111) resultassem em um dígito BCD5211 também inválido,ØØ1Ø, escolhido arbitrariamente. Se não for importante checar a validade da entrada podemos atribuir o valor "XXXX" para as saídas resultantes de códigos BCD8421 inválidos e conseguir alguma simplificação adicional no circuito. Isto fica como exercício para o leitor.

	D	C	B	A	Y_3	Y_2	Y_1	Y_0	
	0	0	0	0	0	0	0	0	
	0	0	0	1	0	0	0	1	
	0	0	1	0	0	0	1	1	
	0	0	1	1	0	1	0	1	
	0	1	0	0	0	1	1	1	
	0	1	0	1	1	0	0	0	
	0	1	1	0	1	0	0	1	
	0	1	1	1	1	0	1	1	
	1	0	0	0	1	1	0	1	
	1	0	0	1	1	1	1	1	
Entradas BCD8421 inválidas	1	0	1	0	0	0	1	0	Saídas BCD5211 inválidas
	1	0	1	1	0	0	1	0	
	1	1	0	0	0	0	1	0	
	1	1	0	1	0	0	1	0	
	1	1	1	0	0	0	1	0	
	1	1	1	1	0	0	1	0	

Fig. 9.10 - Tabela da verdade para decodificador BCD8421 a BCD 5211.

Da tabela da verdade tiramos os mapas de Karnaugh para as saídas; estes mapas, com a simplificação pelos "1"'s e as equações resultantes, estão mostrados na figura 9.11:

DC \ BA	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	0	0	0	0
10	1	1	0	0

$Y_3 = D.\overline{C}.\overline{B}+\overline{D}.C.A+\overline{D}.C.B$

DC \ BA	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

$Y_2 = \overline{D}.\overline{C}.B+D.\overline{C}.\overline{B}$

DC \ BA	00	01	11	10
00	0	0	0	1
01	1	0	1	0
11	1	1	1	1
10	0	1	1	1

$Y_1 = C.\overline{B}.\overline{A}+C.B.A+\overline{C}.B.\overline{A}+$
$+D.C+D.A$

DC \ BA	00	01	11	10
00	0	1	1	1
01	1	0	1	1
11	0	0	0	0
10	1	1	0	0

$Y_0 = \overline{D}.\overline{C}.A+\overline{D}.C.\overline{A}+D.\overline{C}.\overline{B}+\overline{D}.B$

Fig. 9.11 - Tabelas da verdade e simplificação para o decodificador de BCD8421 para BCD5211.

A figura 9.12 ilustra uma possível implementação do circuito, utilizando circuitos integrados CMOS. Nesta figura, adicionamos um detetor de validade, que fornecerá um sinal em nível "1" quando a saída do decodificador for ØØ1Ø (código BCD5211 inválido).

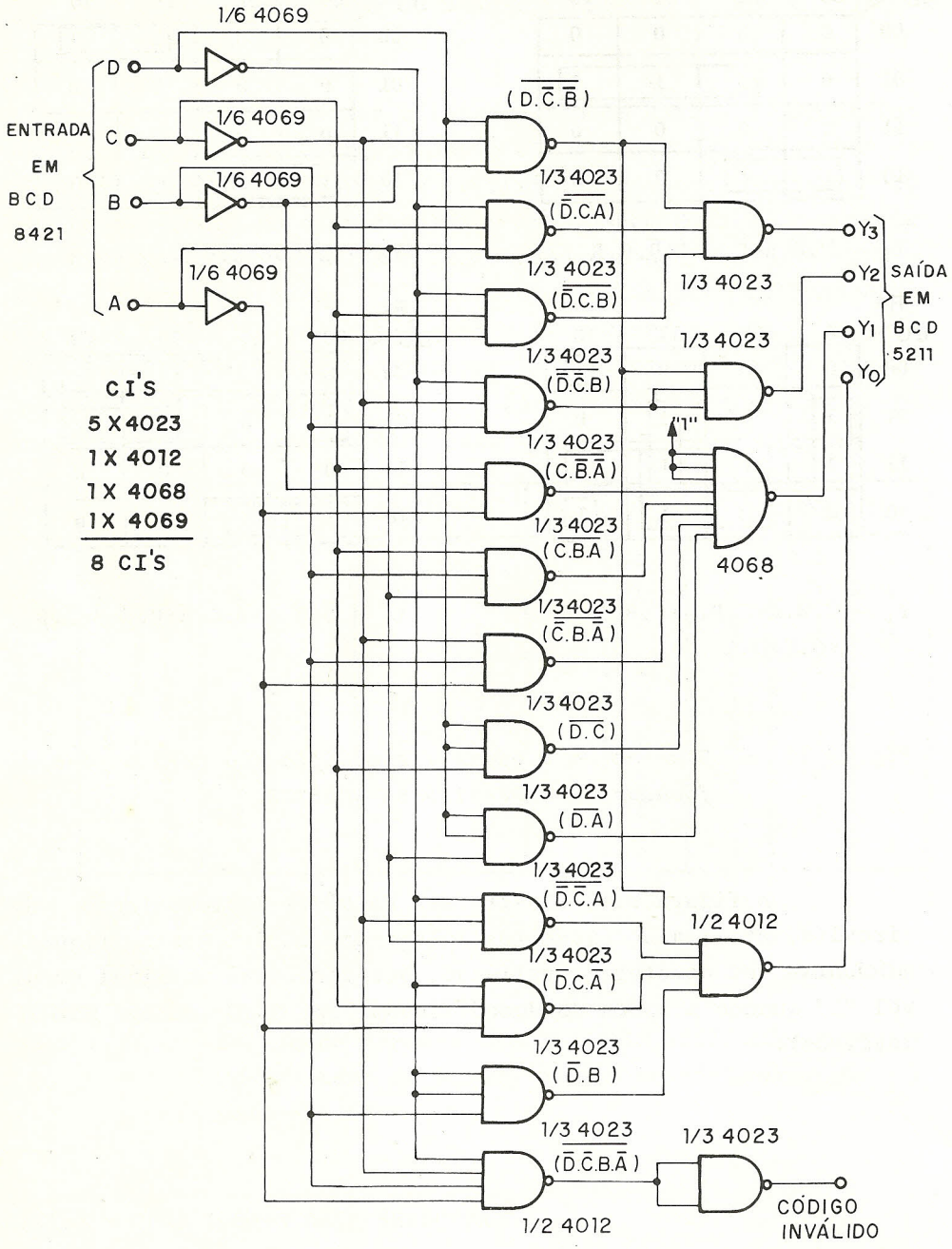
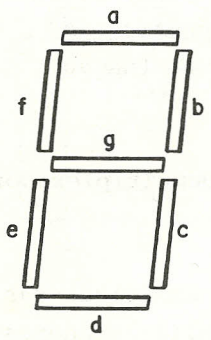


Fig. 9.12 - Decodificador BCD8421 para BCD5211 com deteção de código inválido.

Na prática é rara a necessidade de se projetar um decodificador como o de exemplo anterior. Os decodificadores mais usados estão já disponíveis na forma de CI's. É o caso do decodificador BCD para decimal, Excesso-3 para decimal e outros, que serão vistos em outro item. De qualquer modo, com as técnicas de projeto conhecidas é possível construir qualquer decodificador que seja necessário.

Um dispositivo capaz de mostrar uma informação de uma forma inteligivel para o ser humano denomina-se mostrador ou DISPLAY. Um dos DISPLAYS mais utilizados na atualidade é o chamado DISPLAY de 7 segmentos, mostrado na figura 9.13, juntamente com a tabela de decodificação de BCD para 7 segmentos.



ON = SEGMENTO ACESO
OFF = SEGMENTO APAGADO

DECIMAL	a	b	c	d	e	f	g
0	ON	ON	ON	ON	ON	ON	OFF
1	OFF	ON	ON	OFF	OFF	OFF	OFF
2	ON	ON	OFF	ON	ON	OFF	ON
3	ON	ON	ON	ON	OFF	OFF	ON
4	OFF	ON	ON	OFF	OFF	ON	ON
5	ON	OFF	ON	ON	OFF	ON	ON
6	OFF	OFF	ON	ON	ON	ON	ON
7	ON	ON	ON	OFF	OFF	OFF	OFF
8	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	OFF	OFF	ON	ON

Fig. 9.13 - Display de 7 segmentos e decodificação BCD para 7 segmentos.

Os segmentos podem ser filamentos incandescentes, como na válvula NUMITRON, diodos emissores de luz (FND70, FND357, FND 560, etc) ou cristais líquidos. Dos três tipos, o maior consumo de corrente ocorre no caso de filamentos incandescentes e o menor com os cristais líquidos. O DISPLAY de sete segmentos pode ser do tipo catodo comum ou anodo comum. No tipo catodo comum todos os LED'S (LIGHT EMITTING DIODE - diodo emissor de luz) estão interligados pelo catodo, que vai ao terra; deve-se

aplicar um nível lógico "1" ao segmento para que ele acenda. No tipo anodo comum, todos os LED'S estão interligados pelo anodo, que vai a Vcc; deve-se aplicar um nível lógico "Ø" ao segmento para que ele acenda. Resumindo, na tabela da figura 9.13 devemos considerar:

a- display de catodo comum => ON = nível lógico "1"
OFF = nível lógico "Ø"

b- display de anodo comum => ON = nível lógico "Ø"
OFF = nível lógico "1".

Para construirmos um decodificador BCD/7-segmentos podemos utilizar uma técnica semelhante à aplicada anteriormente. O decodificador terá quatro entradas (os bits do código BCD) e sete saídas (o controle dos segmentos). No entanto, como este decodificador tem uma enorme aplicação na prática já existe uma grande variedade de CI's que realizam esta função.

9.4 - Decodificadores, Multiplexadores e Demultiplexadores da Família TTL

Apresentamos, a seguir, uma breve descrição das características de alguns dos decodificadores, multiplexadores e demultiplexadores da série 74TTL.

7442 - BCD TO DECIMAL DECODER

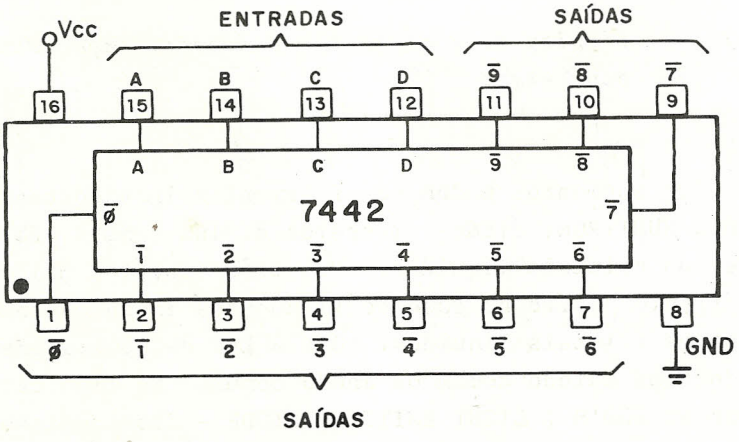


Fig. 9.14 - 7442.

O 7442 é um decodificador de BCD para decimal. De acordo com o valor binário das entradas D, C, B e A (D = MSB), uma das saídas ficará ativa (estado lógico "Ø") enquanto as demais permanecerão inativas (estado lógico "1"). Por exemplo, se D = "Ø", C = "1", B = "Ø" e A = "1", a saída $\overline{5}$ irá para "Ø" e as demais permanecerão em "1". Caso o dígito BCD aplicado às entradas seja inválido (1010 até 1111), nenhuma das saídas ficará ativa. As saídas são TTL normal (I_{OL}=16 mA, I_{OH}=400 µA) e o atraso de propagação típico é de 17 ns.

7445 - BCD TO DECIMAL DECODER/DRIVER
(30 - VOLT, 80 mA OPEN COLLECTOR OUTPUTS)

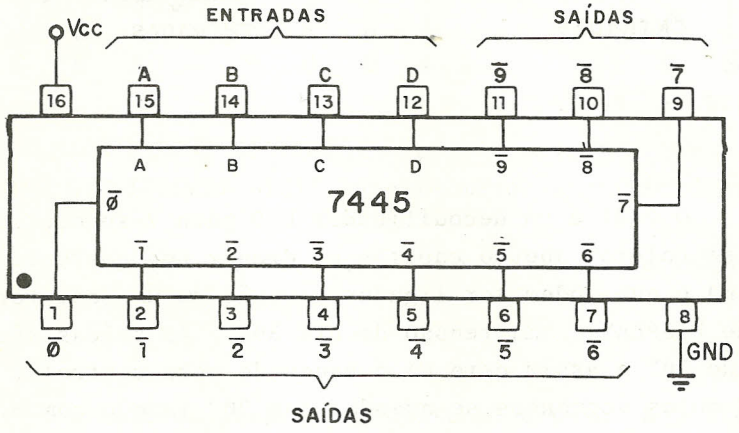


Fig. 9.15 - 7445.

O 7445 é idêntico ao 7442 quanto ao funcionamento. A única diferença reside em que as saídas são do tipo OPEN-COLLECTOR. Quando no estado lógico "Ø", uma saída é capaz de absorver uma corrente I_{OL} de até 80 mA. Para que o estado lógico "1" seja atingido há necessidade de um resistor de PULL-UP externo, que pode estar conectado a uma tensão de até 30V. O atraso de propagação típico é de 45 ns.

7447 - BCD TO 7-SEGMENT DECODER/DRIVER
(30 VOLT, 40 mA, OPEN COLLECTOR OUTPUTS)

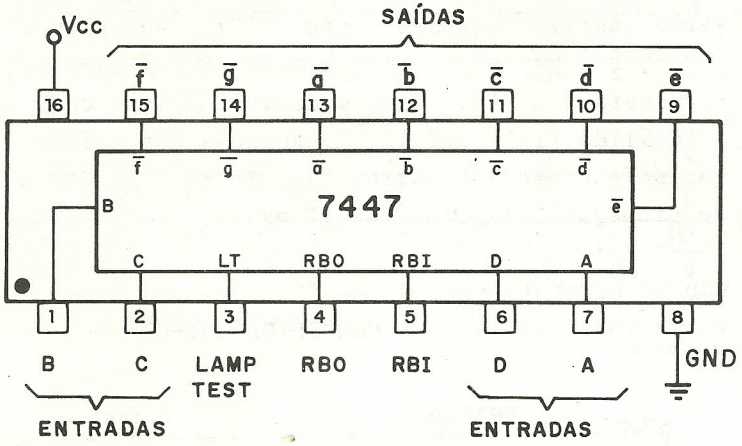


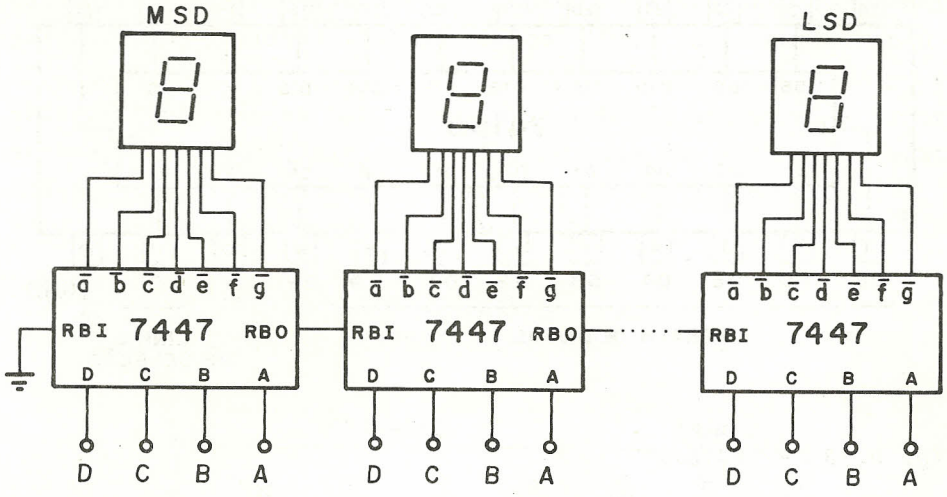
Fig. 9.16 - 7447.

O 7447 é um decodificador BCD para 7-segmentos com saídas em coletor aberto capazes de drenar uma corrente I_{OL} de até 40 mA e que podem ser ligadas através de um resistor ou segmento de DISPLAY a uma tensão de até 30V. As saídas são ativas no estado "Ø" e assim este CI é adequado para controlar um DISPLAY cujos segmentos se acendam com "Ø" (anodo comum, por exemplo). Quando o CI é usado para comandar um DISPLAY de LED'S, resistores externos de limitação de corrente (330Ω tipicamente) devem ser utilizados.

A entrada LT (LAMP TEST) deve permanecer no estado "1" para operação normal; um nível "Ø" nesta entrada força todas as saídas para "Ø" provocando o acendimento de todos os segmentos do DISPLAY.

A entrada RBI (RIPPLE BLANK INPUT) é utilizada para apagar os zeros não significativos de uma série de DISPLAYS. Quando RBI está em "Ø" e D = C = B = A = Ø, todas as saídas vão para "1", apagando o DISPLAY; a saída RBO (RIPPLE BLANKING OUTPUT) vai para "Ø" e pode ser utilizada como RBI para o estágio anterior. A figura 9.17 ilustra este esquema de apagamento dos ze

ros não significativos:



MSD - MOST SIGNIFICANT DIGIT (dígito mais significativo)
LSD - LEAST SIGNIFICANT DIGIT (dígito menos significativo)

Fig. 9.17 - Esquema de ligação para apagamento dos zeros não significativos.

O atraso de propagação deste CI é de 45 ns, tipicamente.

74150 - 1 - OF-16 DATA SELECTOR

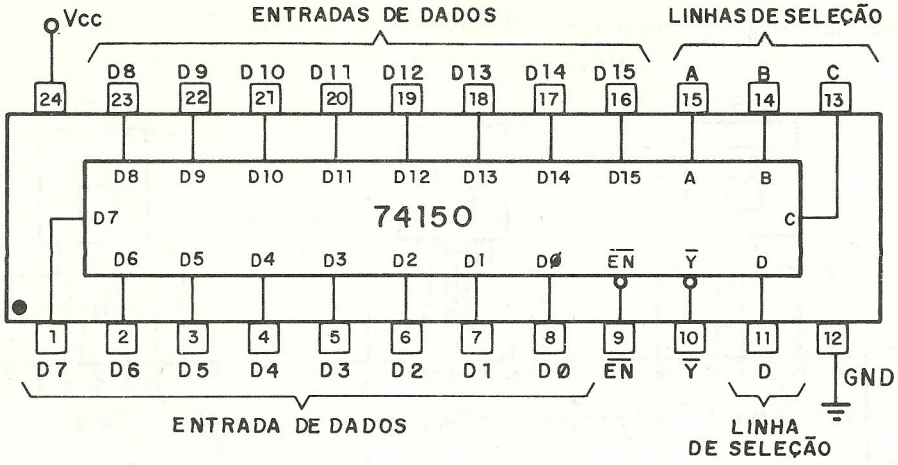


Fig. 9.18 - 74150.

Este CI é um multiplexador com 16 linhas de entrada. Para operação normal, a linha de habilitação $\overline{EN}$ deve ser igual a "Ø", se $\overline{EN}$ está em "1", a saída fica em "1" independente das entradas e das linhas de seleção.

Com $\overline{EN}$ em "Ø", o conteúdo da entrada selecionada pelas linhas D, C, B e A (D = MSB) aparece invertido na saída $\overline{Y}$.

O atraso de propagação é de 23 ns, tipicamente.

74151 - 1-OF-8 DATA SELECTOR

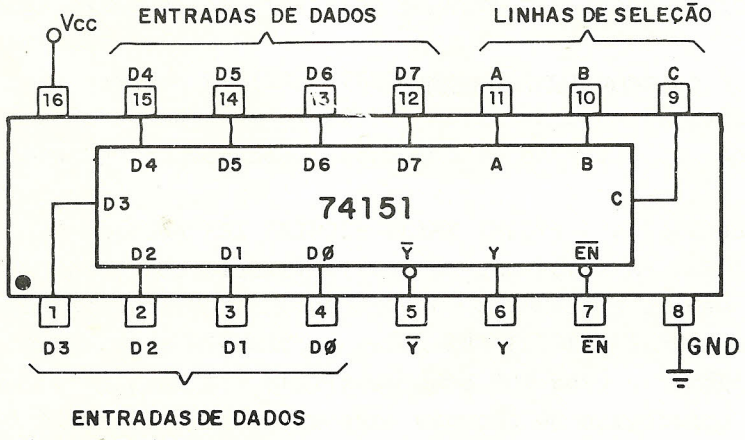


Fig. 9.19 - 74151.

O 74151 é um multiplexador com 8 linhas de entrada e 2 linhas de saída. Para operação normal, a entrada de habilitação $\overline{EN}$ deve estar em "Ø", se $\overline{EN}$ está em "1" a saída Y fica em "Ø" e a saída $\overline{Y}$ fica em "1", independente das entradas e das linhas de seleção.

Com $\overline{EN}$ = "Ø", o conteúdo da entrada selecionada através das linhas C, B e A (C = MSB) aparece invertido na saída $\overline{Y}$ e sem inversão na saída Y.

O atraso é tipicamente de 19 ns.

74153 - DUAL 1-OF-4 DATA SELECTOR

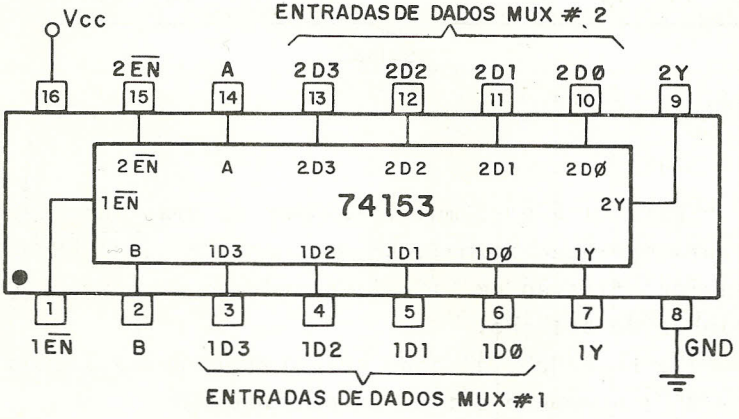


Fig. 9.20 - 74153.

O CI contém dois multiplexadores de 4 entradas. A descrição abaixo vale para ambos.

Para operação normal, a entrada de habilitação $\overline{EN}$ deve ser "Ø"; se $\overline{EN}$ = "1" a saída Y fica em "Ø", independente das entradas de dados e das linhas de seleção.

Com $\overline{EN}$ = "Ø" o conteúdo da entrada de dados selecionada pelas linha B e A (B = MSB) aparece na saída Y.

Observe que as linhas de seleção são comuns aos dois multiplexadores. O atraso tipico é de 44 ns.

74154 - 1-OF-16 DATA DISTRIBUTOR

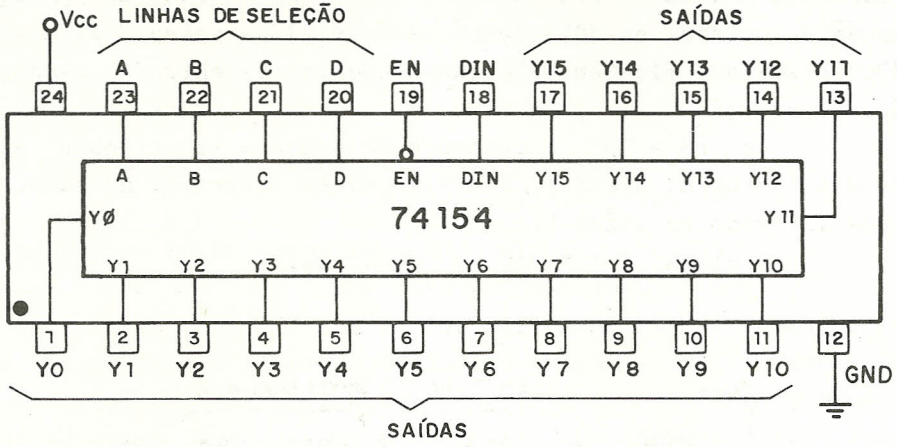


Fig. 9.21 - 74154.

Este CI é um demultiplexador de uma linha para 16 linhas. Para operação normal, $\overline{EN}$ deve ser "Ø" , se $\overline{EN}$ = "1" todas as saídas ficarão em "1", independente da entrada de dados e das linhas de seleção.

Se $\overline{EN}$ = "Ø", a saída selecionada pelas linhas D, C, B e A (D = MSB) assume o estado da entrada DIN.

O atraso típico é de 49 ns.

74155 - DUAL 1-OF-4 DATA DISTRIBUTOR

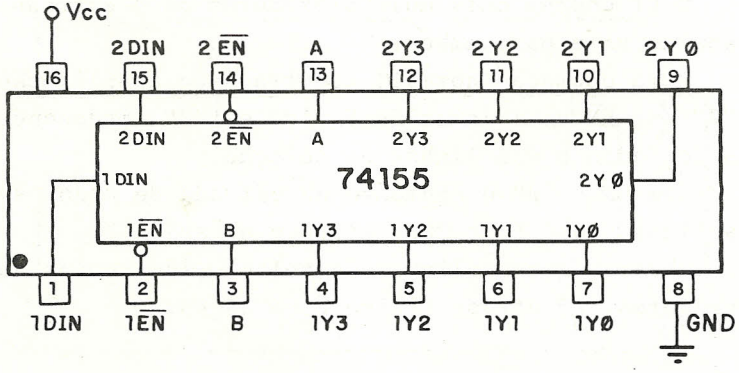


Fig. 9.22 - 74155.

Este CI contém dois demultiplexadores de 1 linha para 4 linhas. A descrição que se segue é válida para ambos.

Para operação normal, a entrada de habilitação $\overline{EN}$ deve estar em "Ø". Se $\overline{EN}$ = "1" as quatro saídas ficam em "1", independente da entrada de dados DIN e das linhas de seleção.

Se $\overline{EN}$ = "Ø", o conteúdo da entrada de dados correspondente é transferido invertido para a saída selecionada pelas linhas B e A (B = MSB), no caso do demultiplexador ≠ 1, e sem inversão no caso do demultiplexador ≠ 2. A figura 9.23 mostra as tabelas da verdade para os dois demultiplexadores:

B	A	$1\overline{EN}$	1DIN	$1Y_\phi$	$1Y_1$	$1Y_2$	$1Y_3$
X	X	1	X	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	1	0	1	1	1
0	1	0	0	1	1	1	1
0	1	0	1	1	0	1	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	0

(a) Tabela da verdade para o demultiplexador ≠ 1.

B	A	$2\overline{EN}$	2DIN	$2Y_\phi$	$2Y_1$	$2Y_2$	$2Y_3$
X	X	1	X	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	1
0	1	0	0	1	0	1	1
0	1	0	1	1	1	1	1
1	0	0	0	1	1	0	1
1	0	0	1	1	1	1	1
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	1

(b) Tabela da verdade para o demultiplexador ≠ 2.

Fig. 9.23 - Tabelas da verdade para o 74155 utilizado como demultiplexador de 1 para 4 dual.

O 74155 pode ser utilizado também como um decodificador de 3 para 8. Para tanto basta conectar as entradas 1DIN e 2DIN externamente, passando a utilizá-las como uma terceira linha de seleção. As entradas $1\overline{EN}$ e $2\overline{EN}$ também devem ser conectadas externamente, passando a atuar como uma habilitação comum ou uma entrada de dados. A tabela da verdade para este modo de operação está ilustrada na figura 9.24. Observe que, se a linha de habilitação ficar sempre em "Ø", temos um decodificador de 3 para 8, se um sinal variável for aplicado a esta linha, teremos um demultiplexador de 1 para 8.

C	B	A	$\overline{EN}$	$2Y_\phi$	$2Y_1$	$2Y_2$	$2Y_3$	$1Y_\phi$	$1Y_1$	$1Y_2$	$1Y_3$
0	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	0	1	1	1	1	1	1
0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	0	1	1	1	0	1	1	1	1
1	0	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	1	1	0	1
1	1	1	0	1	1	1	1	1	1	1	0
X	X	X	1	1	1	1	1	1	1	1	1

C => 1DIN e 2DIN conectados juntos
$\overline{EN}$ => $1\overline{EN}$ e $2\overline{EN}$ conectados juntos

Fig. 9.24 - Tabela da verdade para o 74155 utilizado como decodificador de 3 para 8 ou demultiplexador de 1 para 8.

74157 - QUAD 1-OF-2 DATA SELECTOR

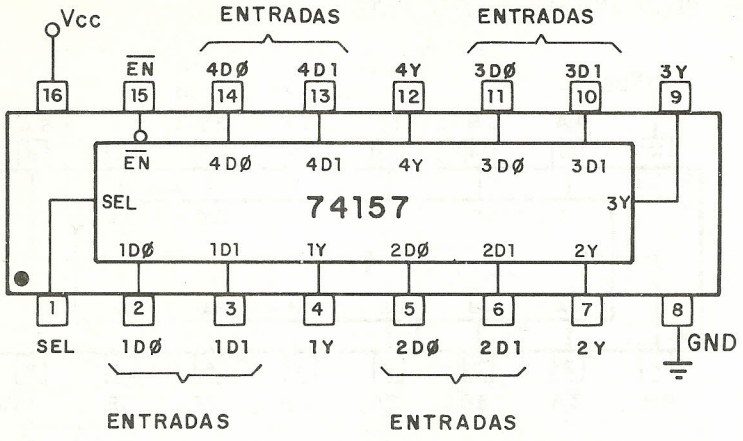


Fig. 9.25 - 74157.

O 74157 contém quatro multiplexadores de 2 linhas para 1 linha. A descrição que se segue é válida para todos eles.

Para operação normal, $\overline{EN}$ deve ser igual a "Ø", se $\overline{EN}$= "1", todas as saídas ficarão em "Ø" independente das entradas e da linha de seleção.

Com $\overline{EN}$ = "Ø" o conteúdo da entrada selecionada pela linha SEL é transferido para a saída. Observe que as linhas de habilitação e seleção são comuns aos quatro multiplexadores. O atraso é de 18 ns, tipicamente.

9.5 - Decodificadores, Multiplexadores e Demultiplexadores da Família CMOS

Também na série 4ØØØ CMOS encontramos uma grande variedade de decodificadores, multiplexadores e demultiplexadores. A seguir, apresentamos as características de alguns deles.

4019 - QUAD AND/OR SELECT GATE

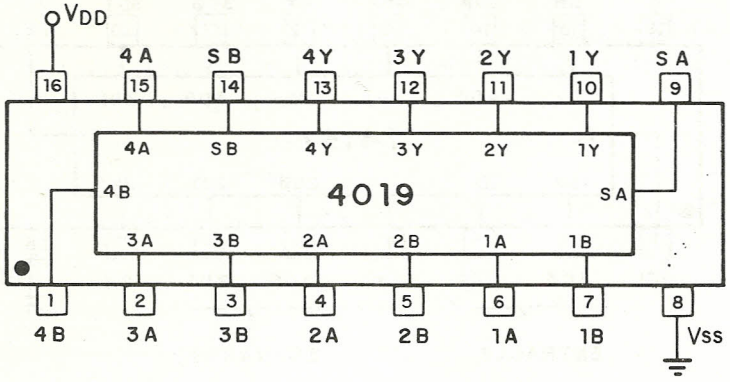


Fig. 9.26 - 4019.

Este CI é usualmente utilizado como um multiplexador de duas linhas para uma linha.

Os quatro blocos AND/OR que o compõem obedecem a tabela da verdade apresentada na figura 9.27:

SB	SA	Y
0	0	0
0	1	A
1	0	B
1	1	A+B

Fig. 9.27 - Tabela da verdade para o 4019.

Esta tabela nos mostra que o 4019 pode realmente ser utilizado como um multiplexador. Se SA = "1" e SB = "Ø", a entrada A é selecionada e se SA = "Ø" e SB = "1", a entrada B é selecionada. Com SA = SB = "Ø" a saída permanece em "Ø" e com SA = SB = "1" a saída é o resultado da operação OU entre as entradas A e B. Isto se aplica aos quatro elementos contidos no CI, pois as linhas SA e SB são comuns. O atraso é 35 ns para V_{DD} = 10V e 200 ns para V_{dd} = 5V.

4028 - BCD TO DECIMAL DECODER

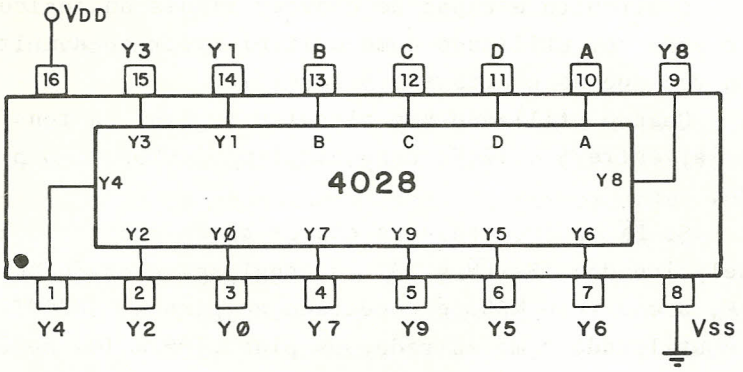


Fig. 9.28 - 4028.

O 4028 é um decodificador de BCD para decimal. de acordo com o valor binário das entradas D, C, B e A (D = MSB) uma e somente uma das saídas irá para "1". Por exemplo, se D = "Ø", C = "1", B = "1" e A = "Ø", a saída Y_6 ficará em "1" e as demais permanecerão em "Ø". Se o dígito BCD na entrada for inválido (1Ø1Ø até 1111) todas as saídas permanecerão em "Ø".

O atraso é de 100 ns para V_{DD} = 10V e de 250 ns para V_{DD} = 5V.

4051 - 1-OF-8 SWITCH

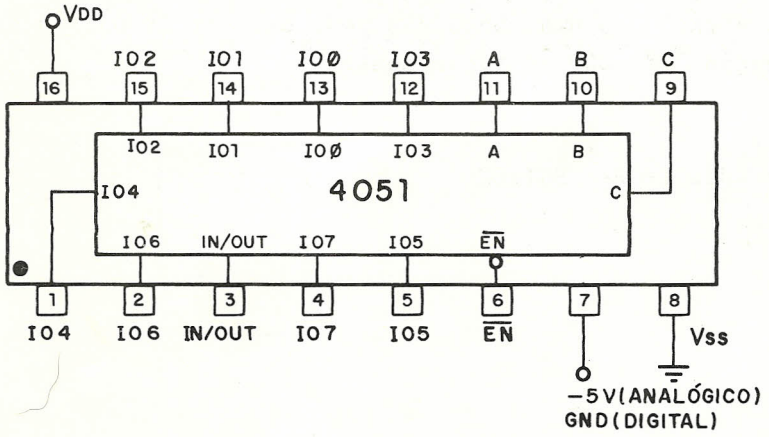


Fig. 9.29 - 4051.

Este CI, juntamente com o 4052, 4053 e 4067, é um componente tipicamente CMOS, que não encontra paralelo em TTL.

O circuito é capaz de chavear sinais analógicos ou digitais e pode ser utilizado como multiplexador ou demultiplexador, uma vez que as chaves são bilaterais.

Quando utilizado num circuito digital, a tensão V_{DD} pode variar entre 5 e 15 V, como usual para CMOS, e o pino 7 é aterrado.

Se $\overline{EN}$ = "1", todas as chaves estão abertas e nenhum canal é selecionado. Se $\overline{EN}$ = "Ø", o canal selecionado pelas entradas C, B e A (C = MSB) é conectado ao pino 3, IN/OUT. Se o pino 3 é utilizado como entrada, os pinos IOØ a IO7 serão as saídas e o circuito se comportará como um demultiplexador de 1 linha para 8 linhas. Se o pino 3 é utilizado como saída, os pinos IOØ a IO7 serão as entradas e o circuito se comportará como um multiplexador de 8 linhas para 1 linha.

Quando utilizado para chaveamento de sinais analógicos, V_{DD} = 5V e o pino 7 deve ser conectado a -5V. Os sinais de seleção, C, B e A, podem variar de Ø (nível Ø) a 5V (nível "1"). Se $\overline{EN}$ = "1", nenhum canal é selecionado; se $\overline{EN}$ = "Ø", a chave selecionada pelas entradas C, B e A se fecha. O circuito pode ser usado como multiplexador ou demultiplexador e os sinais analógicos podem variar entre +5 e -5V.

Em qualquer caso, as chaves abertas são circuitos abertos e fechadas representam uma resistência de 120Ω. A resistência de carga mínima para qualquer saída é de 100Ω e a corrente máxima através de uma chave não pode exceder 25 mA. O atraso de propagação é de 200 ns tipicamente.

4052 - DUAL 1-OF-4 SWITCH

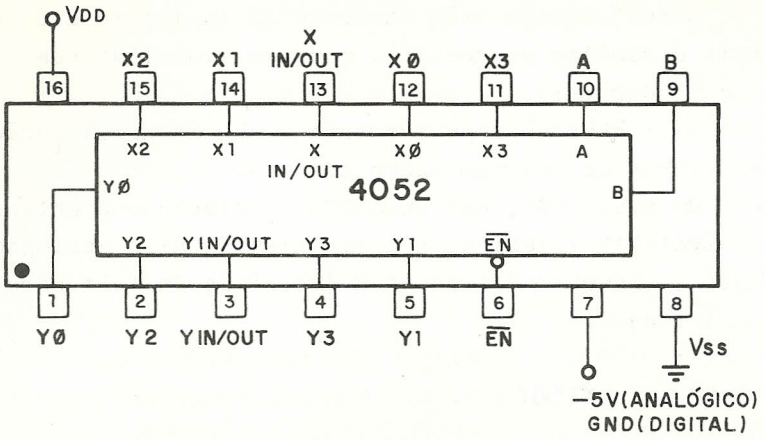


Fig. 9.30 - 4052.

O 4052 contém duas chaves CMOS de 1 para 4 e seu funcionamento é similar ao do 4051. As entradas B e A (B = MSB)são a seleção comum para as duas seções. Refira-se à descrição do 4051 para informações adicionais.

4053 - TRIPLE 1-OF-2 SWITCH

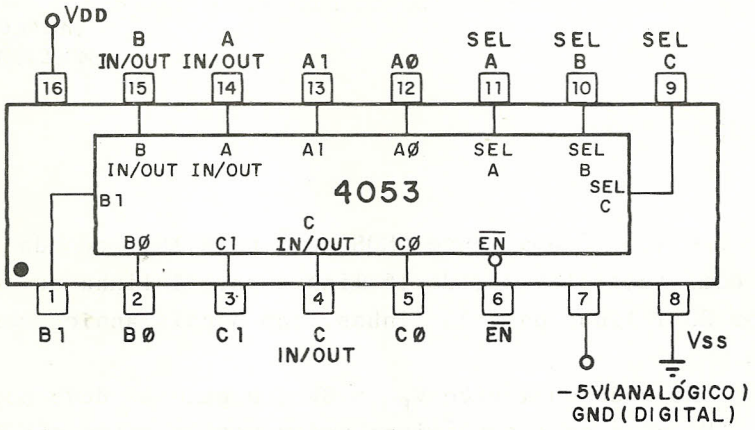


Fig. 9.31 - 4053.

O 4053 contém três chaves CMOS de 1 para 2 e seu funcionamento é similar ao do 4051, podendo trabalhar com sinais analógicos e digitais.

As três seções possuem uma linha de habilitação comum, mas linhas de seleção independentes.

Um nível "Ø", na linha SELA, seleciona a entrada/saída AØ e um nível "1" seleciona a entrada/saída A1 e analogamente para as outras chaves. Refira-se à descrição do 4051 para informações adicionais.

4067 - 1-OF-16 SWITCH

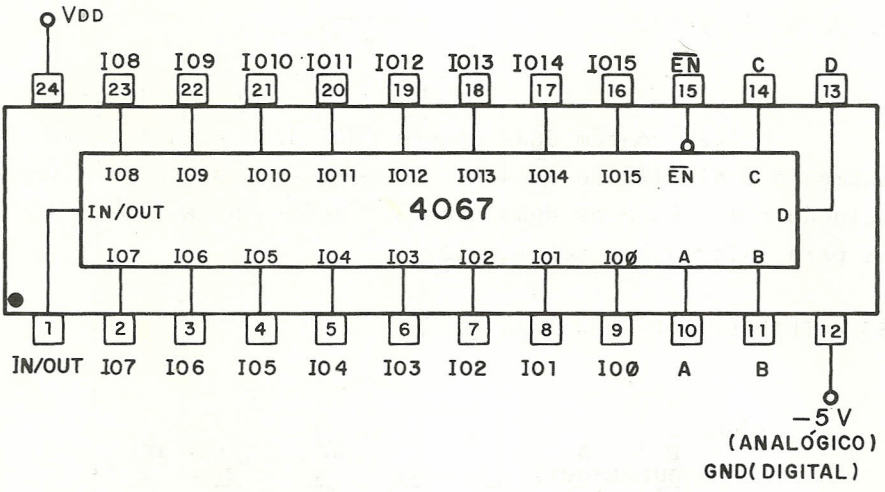


Fig. 9.32 - 4067.

O 4067 é uma chave CMOS de 1 para 16, podendo ser utilizado como multiplexador de 16 linhas para 1 linha ou demultiplexador de 1 linha para 16 linhas, com sinais analógicos ou digitais.

No modo analógico V_{DD} = 5V e o pino 12 deve ser conectado a -5V, os sinais analógicos podem variar entre +5V e -5V.

No modo digital V_{DD} pode variar entre +3V e + 15V e o pino 12 deve ser aterrado.

A resistência ON é de 200Ω e o atraso de propagação é de 200 ns para V_{DD} = 10V e 400 ns para V_{DD} = 5V. Refira-se à descrição do 4051 para informações adicionais.

4026 - DECADE COUNTER WITH 7-SEGMENT DECODED OUTPUT (SYNCHRONOUS)

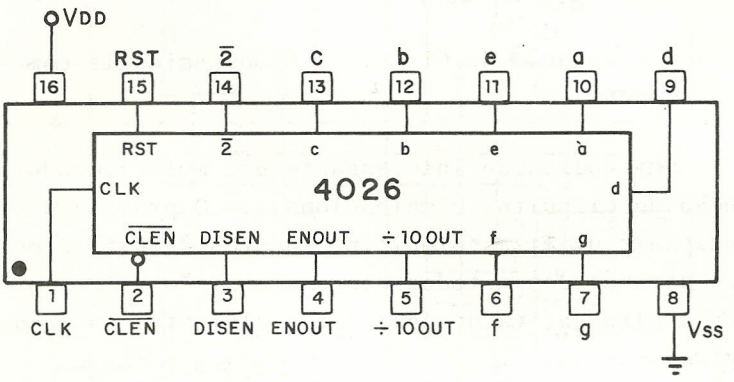


Fig. 9.33 - 4026.

Este CI é um contador síncrono divisor por 10, ou década, que já engloba um decodificador BCD para 7 segmentos.

Para operação normal, RST deve ser "Ø" e $\overline{CLEN}$ deve ser "Ø". Um nível "1" em RST resseta o contador para "Ø" e impede a contagem. Um nível "1" em $\overline{CLEN}$ (CLOCK ENABLE) inibe a aplicação do CLOCK.

O contador é gatilhado na transição positiva do CLOCK. Uma onda quadrada com 1/10 da freqüência do CLOCK está disponível no pino 5; o pino 14 fornece um sinal que permanece em "1" a maior parte do tempo e vai para "Ø" quando a contagem é igual a ØØ1Ø.

A linha DISEN controla a habilitação do DISPLAY. Esta linha deve ser igual a "1" para operação normal; se DISEN = "Ø", as saídas de a até f permanecem inativas, em "Ø".

Este CI é adequado para controlar DISPLAYS que acendam com o nível lógico "1" em seus segmentos. A corrente máxima que pode ser fornecida a um segmento é 1,2 mA para V_{DD} = 5V

e 5 mA para V_{DD} = 10V.

A saída ENOUT segue a linha DISEN.

A freqüência máxima do CLOCK é de 5MHZ para V_{DD} = 10V e 2,5 MHZ para V_{DD} = 5V.

9.6 - Aplicações de Decodificadores, Multiplexadores e Demultiplexadores

9.6.1 - Implementação de Circuitos Combinacionais com Multiplexadores.

Uma aplicação interessante dos multiplexadores é na construção de circuitos combinacionais. O projeto de circuitos combinacionais utilizando multiplexadores é muitíssimo mais simples que através dos métodos convencionais e vale a pena ser considerada a alternativa de usar estes elementos no lugar de portas lógicas.

Consideremos a função lógica cuja tabela da verdade está ilustrada na figura 9.34:

D	C	B	A	Y	AUX
Ø	Ø	Ø	Ø	1	$\overline{D}$
Ø	Ø	Ø	1	1	1
Ø	Ø	1	Ø	Ø	Ø
Ø	Ø	1	1	Ø	D
Ø	1	Ø	Ø	1	1
Ø	1	Ø	1	1	$\overline{D}$
Ø	1	1	Ø	Ø	Ø
Ø	1	1	1	1	1
1	Ø	Ø	Ø	Ø	$\overline{D}$
1	Ø	Ø	1	1	1
1	Ø	1	Ø	Ø	Ø
1	Ø	1	1	1	D
1	1	Ø	Ø	1	1
1	1	Ø	1	Ø	$\overline{D}$
1	1	1	Ø	Ø	Ø
1	1	1	1	1	1

Fig. 9.34 - Tabela da Verdade de uma função lógica que se deseja implementar.

O projeto do circuito combinacional que realiza a função da figura 9.34 envolve a plotagem da função sobre o mapa de Karnaugh e a minimização das expressões lógicas, o que é um processo relativamente trabalhoso. A função lógica pode ser direta e imediatamente implementada utilizando um multiplexador de 16 linhas para 1 linha, conforme ilustra a figura 9.35:

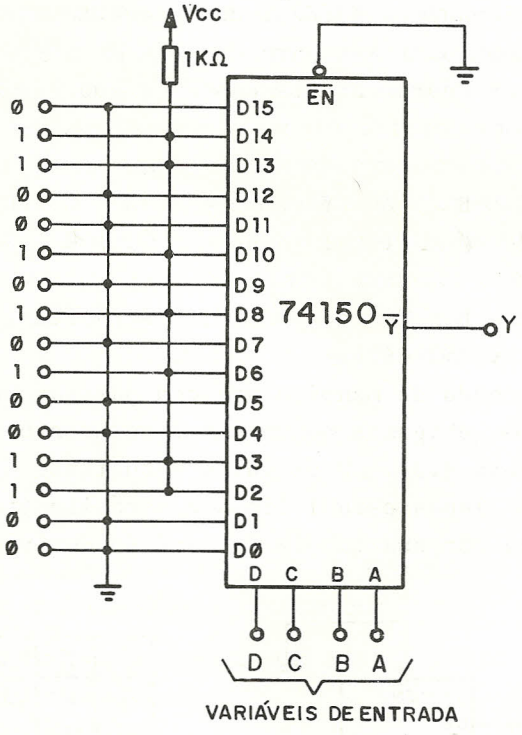


Fig. 9.35 - Função Lógica implementada com multiplexador de 16 linhas para 1 linha.

Aqui as variáveis de entrada foram utilizadas como linhas de seleção do multiplexador. Nas entradas de dados colocou-se o complemento do valor desejado na saída, uma vez que o 74150 fornece a saída invertida em relação à entrada selecionada.

A implementação da figura 9.35 é direta, mas é possí

vel conseguir uma solução melhor e de menor custo. Consideremos apenas as três variáveis de entrada C, B e A e examinemos os dois casos em que C = B = A = "Ø" na tabela da verdade. Temos:

D	C	B	A	Y
Ø	Ø	Ø	Ø	1

D	C	B	A	Y
1	Ø	Ø	Ø	Ø

Relacionando a saída Y com a quarta variável de entrada, D, observamos que Y = $\overline{D}$ para C = B = A = "Ø".

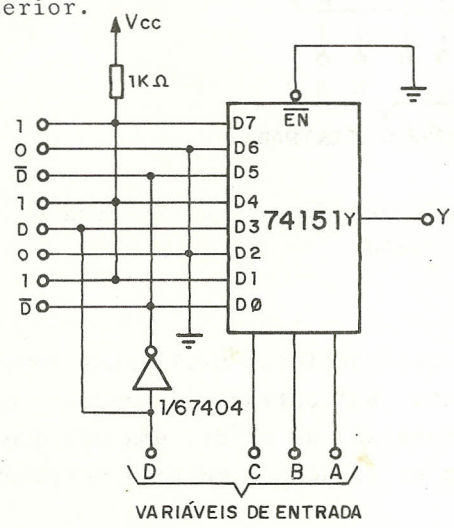
Vejamos agora os dois casos em que C = B = "Ø" e A = "1". Temos:

D	C	B	A	Y
Ø	Ø	Ø	1	1

D	C	B	A	Y
1	Ø	Ø	1	1

ou seja, para C = B = "Ø" e A = "1", Y será sempre igual a "1", independente da variável D.

Procedendo de maneira análoga para as outras combinações de C, B e A, chegamos ao resultado expresso na coluna AUX, mostrada na figura 9.34. O circuito resultante utiliza um multiplexador de 8 linhas para 1 linha e está ilustrado na figura 9.36, juntamente com sua tabela da verdade derivada da análise anterior.



C	B	A	Y
Ø	Ø	Ø	$\overline{D}$
Ø	Ø	1	1
Ø	1	Ø	Ø
Ø	1	1	D
1	Ø	Ø	1
1	Ø	1	$\overline{D}$
1	1	Ø	Ø
1	1	1	1

Fig. 9.36 - Função lógica implementada com multiplexador de 8 linhas para 1 linha.

A construção de circuitos combinacionais utilizando multiplexadores possui diversas vantagens. A que se vê imediatamente é a facilidade do projeto, que é quase a transposição direta da tabela da verdade para as entradas do multiplexador. Outra vantagem, importante em um projeto real, é a facilidade de modificação do circuito. Suponha que, após implementarmos a função da figura 9.34 com portas lógicas, descobríssemos que uma ou duas linhas da tabela devessem ser alteradas. Seria necessário refazer todo o projeto, construindo novamente o mapa de Karnaugh, verificando as novas adjacências, etc. No caso do circuito com o multiplexador bastaria alterar uma ou duas entradas do CI. Embora os multiplexadores possuam um custo mais elevado que as portas lógicas, geralmente o projeto resulta num número menor de CI's, simplificando o lay-out, reduzindo o consumo e, em certos casos, possibilitando menores atrasos. Por todas estas razões, a implementação com multiplexadores é uma alternativa que deve ser examinada e, em muitos casos, preferida ao método convencional.

A técnica de projeto pode ser assim, sistematizada:

a) para um circuito com n variáveis de entrada, é necessário um multiplexador com 2^{n-1} entradas;

b) separe uma variável de entrada, digamos X, e considere as combinações das N-1 variáveis restantes;

c) para cada combinação destas n-1 variáveis de entrada, a saída Y pode ser expressa como: "Ø", "1", X ou $\overline{X}$. Construa a tabela auxiliar para a saída;

d) implemente a função utilizando as n-1 variáveis como linhas de seleção do multiplexador e colocando nas entradas do mesmo os valores obtidos no item anterior.

Para ilustrar a aplicação deste método, vamos refazer o projeto do decodificador BCD8421 para BCD5211 apresentado no item 9.3 . Sigamos os passos a, b, c e d apresentados.

a) Como são quatro as variáveis de entrada, necessitamos de multiplexadores de 8 (2^3) entradas.

b) Arbitrariamente, separemos a variável D e consideremos as combinações das variáveis C, B e A.

c) Por simples inspeção da tabela da verdade da figura 9.10, construímos a tabela auxiliar ilustrada na figura 9.37:

C	B	A	Y_3	Y_2	Y_1	Y_0
0	0	0	D	D	Ø	D
0	0	1	D	D	D	1
0	1	0	Ø	Ø	1	$\overline{D}$
0	1	1	Ø	$\overline{D}$	D	$\overline{D}$
1	0	0	Ø	$\overline{D}$	1	$\overline{D}$
1	0	1	$\overline{D}$	Ø	D	Ø
1	1	0	$\overline{D}$	Ø	D	$\overline{D}$
1	1	1	$\overline{D}$	Ø	1	$\overline{D}$

Fig. 9.37 - Tabela auxiliar.

d) Implementamos o circuito utilizando CI's CMOS; a deteção de código inválido foi feita com portas lógicas. O circuito está ilustrado na figura 9.38:

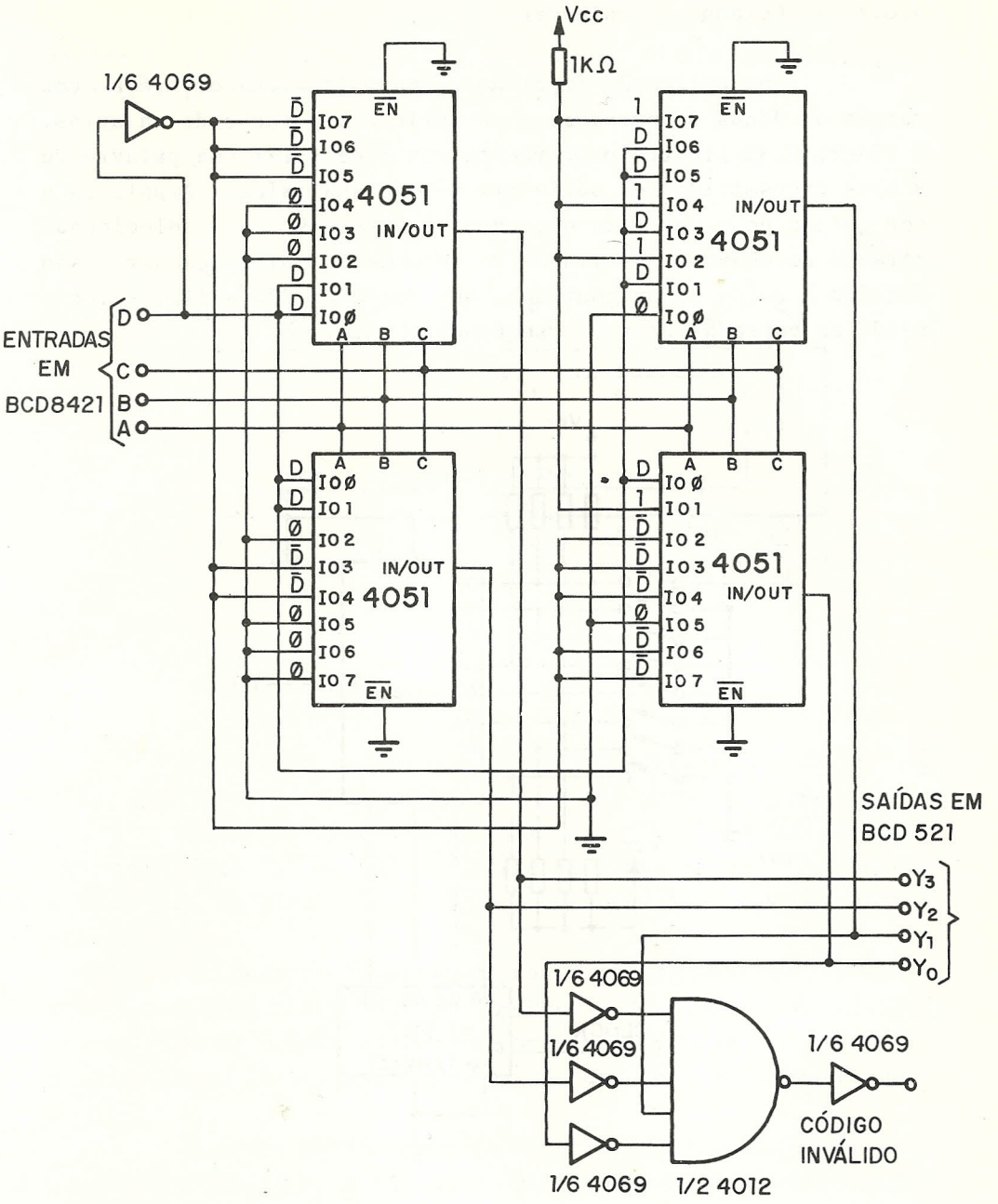


Fig. 9.38 - Decodificador BCD8421 para BCD5211 com deteção de código inválido (implementaçãc com multiplexer).

9.6.2 - Gerador de Palavras

Uma aplicação importante dos multiplexadores é a conversão de dados de paralelo para série e a geração de palavras. A figura 9.39 ilustra um circuito capaz de gerar uma palavra de 8 bits transmitidos em série através de sua saída. A palavra a ser gerada pode ser fixa ou, como na figura 9.39, selecionada através de chaves. As linhas de seleção do multiplexador são ligadas à saída de um contador, que conta de ØØØ a 111, selecionando as entradas DØ a D7, seqüencialmente.

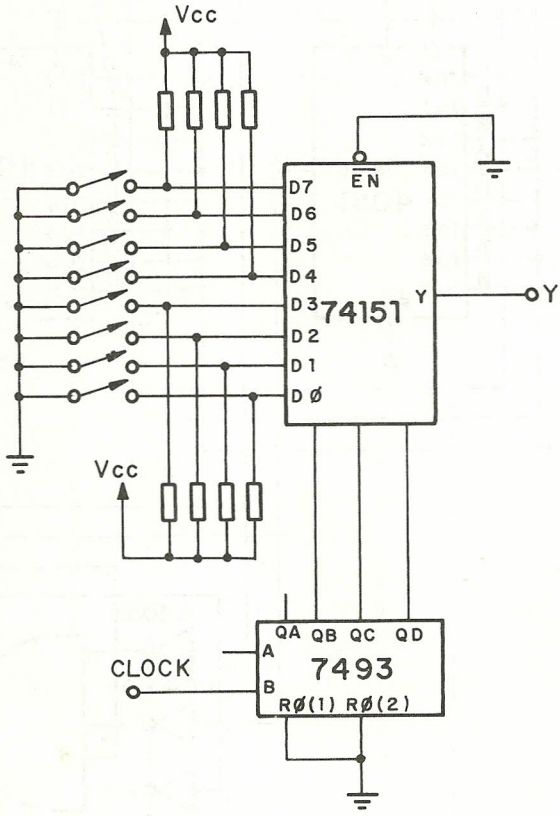


Fig. 9.39 - Gerador de palavras.

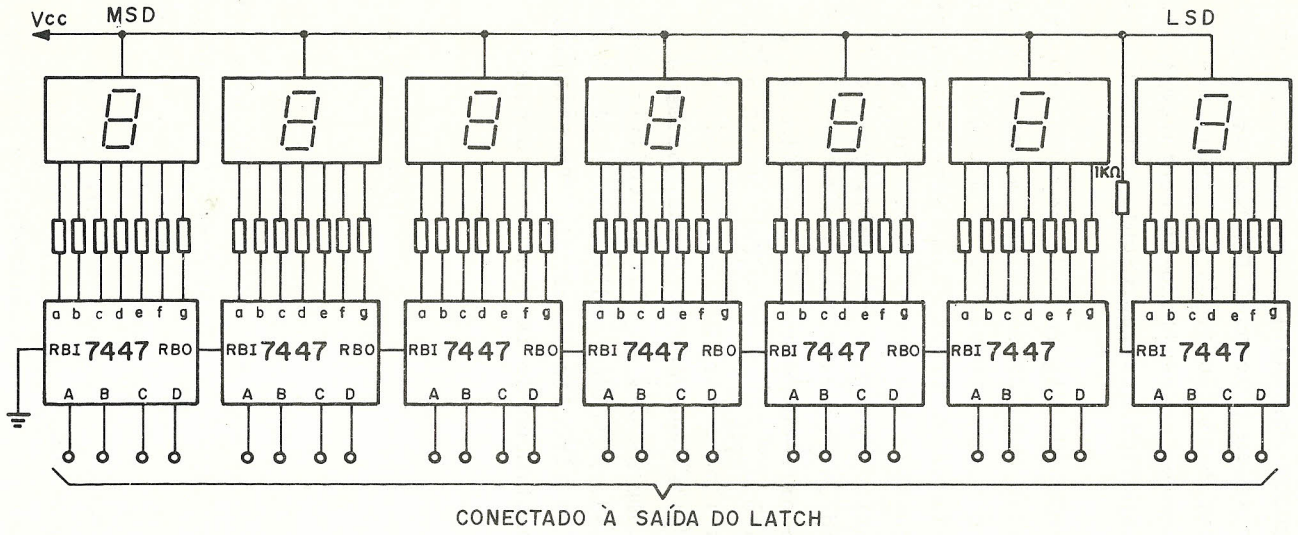
9.6.3 - Mostragem de Resultados

Uma aplicação óbvia dos decodificadores é na mostragem de resultados codificados em binário. No item 7.7 vimos a construção da parte de contagem e geração de base de tempo de um freqüencímetro digital, que está ilustrada na figura 7.46. Naquele circuito, o LATCH contém o resultado da contagem feita pelo contador de pulsos, resultado que representa a freqüência que se deseja medir.

Para que o circuito da figura 7.46 tenha alguma utilidade, é necessário converter a saída do LATCH em uma informação visível. Esta conversão é feita através de um decodificador e DISPLAY. No caso, se utilizarmos DISPLAYS de sete segmentos, necessitaremos de decodificadores BCD para 7-segmentos, como o 7447. A figura 9.40 ilustra a parte de decodificação e DISPLAY que completaria o circuito da figura 7.46.

O arranjo contador/LATCH/decodificador/DISPLAY é encontrado em muitas aplicações. No exemplo do freqüencímetro, as quatro funções são realizadas por CI's separados.

No entanto, há CI's que englobam duas ou mais funções. Há, por exemplo, combinações de LATCH/decodificador, decodificador/DISPLAY, contador/LATCH/decodificador, LATCH/decodificador/DISPLAY em um único circuito integrado, facilitando sobremaneira o projeto e construção de instrumentos digitais.



OBS:. Todos os resistores de 330Ω, com função de limitar a corrente pelos segmentos
Os DISPLAYS são do tipo anodo comum, constituídos de LED's
O circuito prove o apagamento de zeros não significativos.

Fig. 9.40 - Decodificado e DISPLAY para o circuito do freqüencímetro da figura 7.46.

CAPÍTULO 10 - ELEMENTOS ARITMÉTICOS

10.1 - Introdução

Nos sistemas digitais é muito comum a necessidade de realizar operações aritméticas com as informações binárias. Este seria o caso de um computador digital, por exemplo, onde uma das principais finalidades é justamente a realização de operações aritméticas (e lógicas) com as informações que recebe.

Neste capítulo, veremos alguns tópicos sobre aritmética binária e alguns circuitos integrados capazes de realizar operações aritméticas com as informações presentes em suas entradas.

10.2 - Adição de Números Binários

Sejam A e B dois números binários que desejamos somar Há apenas quatro possibilidades:

a) B = "Ø", A = "Ø" - a soma S é igual a "Ø";
b) B = "Ø", A = "1" - a soma S é igual a "1";
c) B = "1", A = "Ø" - a soma S é igual a "1";
d) B = "1", A = "1" - a soma S é igual a "Ø" e vai um; este bit é chamado de CARRY.

A figura 10.1 mostra a tabela da verdade para a soma de dois bits e o circuito lógico capaz de realizá-la. "S" é a soma e "C" o CARRY ou vai-um.

B	A	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \overline{B} \cdot A + B \cdot \overline{A} = A \oplus B$

$C = A \cdot B$

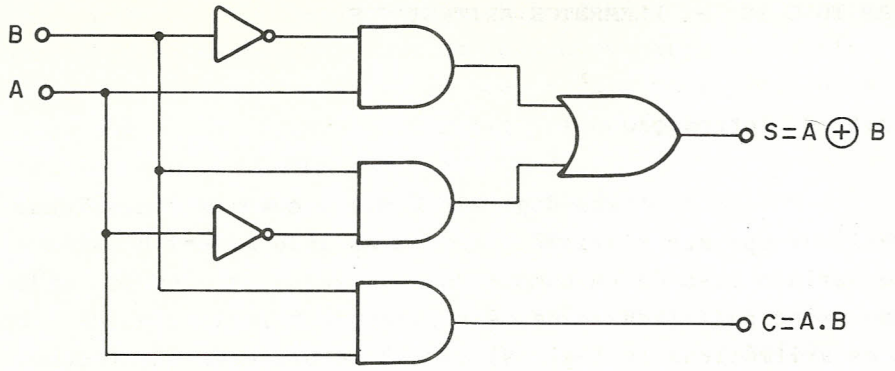


Fig. 10.1 - Tabela da verdade e circuito lógico do meio- somador (HALF-ADDER).

O circuito da figura 10.1, recebe o nome de meio-somador ou HALF-ADDER.

Quando desejamos somar números binários de mais de um bit procedemos a soma bit a bit da direita para a esquerda, a exemplo do que fazemos usualmente com números decimais. Em cada soma de dois bits consideramos também o vai-um ou CARRY proveniente da soma dos bits anteriores. Na figura 10.2 está ilustrada a soma de dois números binários de 8 bits, mostrando de forma bem clara este processo.

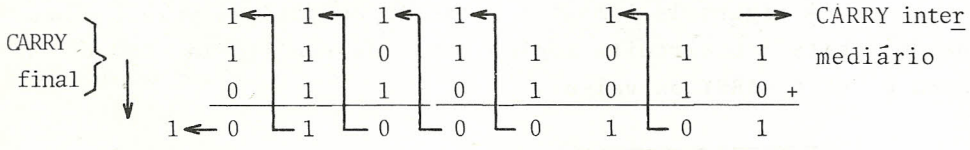


Fig. 10.2 - Adição de dois números binários de 8 bits.

De um modo geral, quando desejamos somar dois bits, An e Bn, quaisquer de dois números binários, devemos levar em consideração o CARRY, vindo do estágio anterior. O resultado será uma soma, Sn, e um CARRY Cn. A figura 10.3 ilustra a tabela

da verdade para esta operação e o circuito lógico capaz de executá-la. Este circuito se denomina somador completo (FULL-ADDER).

C_{n-1}	Bn	An	Sn	Cn
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C_{n-1} - CARRY do estágio n-1
Bn - bit de ordem n de B
An - bit de ordem n de A
Sn - bit de ordem n da soma
Cn - CARRY do estágio n

$$Sn = C_{n-1}.Bn.An + C_{n-1}.\overline{Bn}.\overline{An} + \overline{C}_{n-1}.Bn.\overline{An} + \overline{C}_{n-1}.\overline{Bn}.An$$

$$= C_{n-1}.(Bn.An + \overline{Bn}.\overline{An}) + \overline{C}_{n-1}.(Bn.\overline{An} + \overline{Bn}.An)$$

$$= C_{n-1}.\overline{(Bn \oplus An)} + \overline{C}_{n-1}.(Bn \oplus An)$$

$$= C_{n-1} \oplus (Bn \oplus An)$$

$$Cn = C_{n-1}.Bn + C_{n-1}.An + Bn.An$$

$$= C_{n-1}.(Bn + An) + Bn.An$$

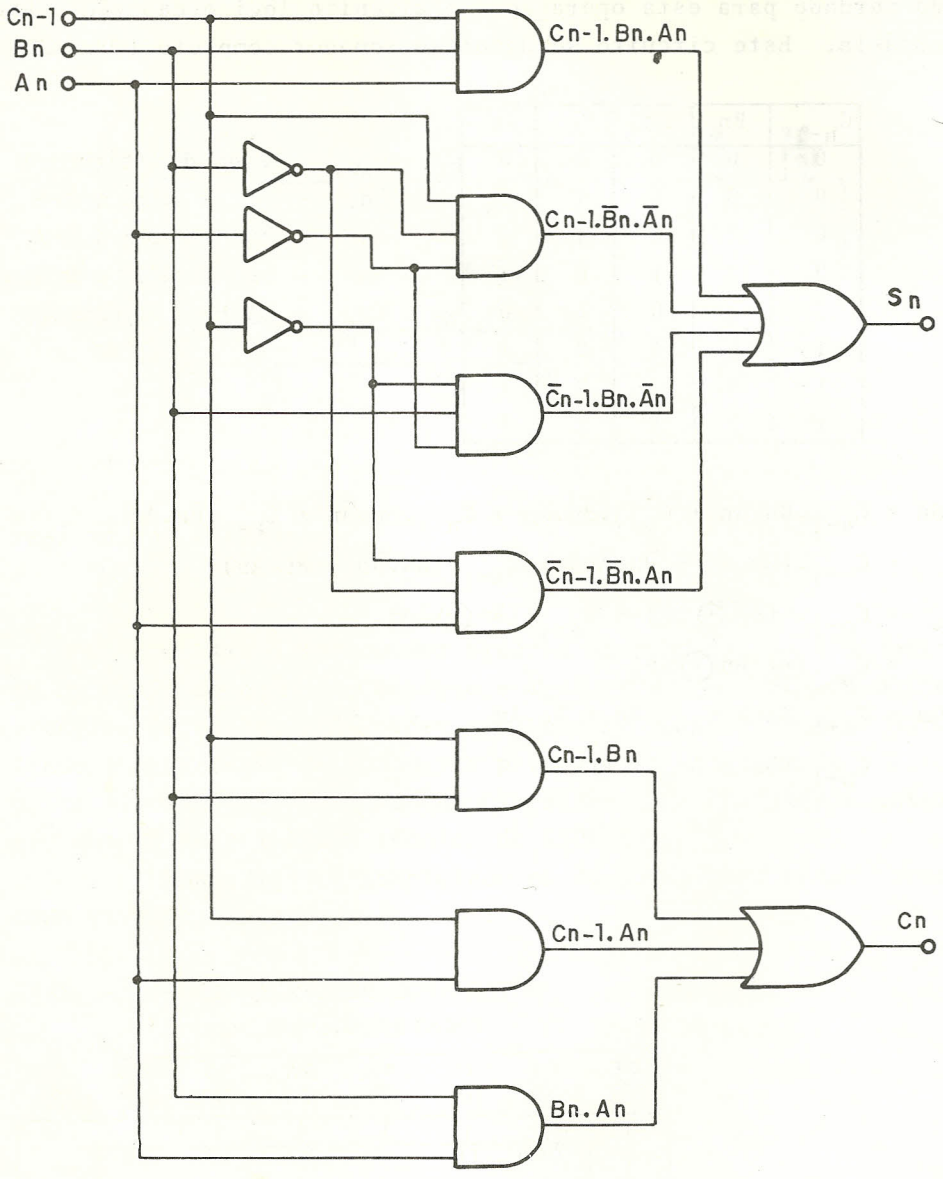


Fig. 10.3 - Tabela da verdade e circuito lógico do somador completo.

Com o somador completo podemos realizar a adição de números binários de n bits. A figura 10.4 ilustra um somador para dois números binários de 4 bits. Cada bloco marcado FA representa um somador completo, como o ilustrado na figura 10.3. Note que o primeiro estágio, que realiza a soma dos bits menos significativos, AØ e BØ, pode ser substituído por um meio-somador como o da figura 10.1; na figura 10.3 usamos um somador completo com a entrada C_{n-1} aterrada.

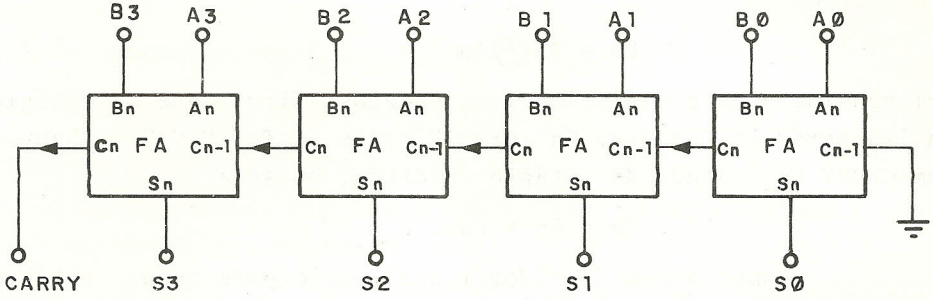


Fig. 10.4 - Somador binário para quatro bits.

O arranjo da figura 10.4 pode ser extendido para qualquer número de bits. Entretanto, observe que o CARRY deve-se propagar através de todos os estágios para que a soma possa ser considerada completa. À medida que o número de estágios aumenta, cresce também o tempo necessário para realizar a adição, o que em certos casos pode ser problemático.

Imagine que cada somador necessite de 30 ns para gerar o seu CARRY. Como a soma de An e Bn só pode ser feita após C_{n-1} estar estável, vai se criando um atraso entre o instante da aplicação das entradas e o momento em que as saídas estão estáveis. Se tivéssemos 20 bits, o atraso seria de 600 ns, o que é excessivo em muitas aplicações.

Observando atentamente a tabela da figura 10.3, podemos tirar algumas conclusões interessantes:

a) se Bn = An = "1", Cn = "1", independente do CARRY C_{n-1} que vem do estágio anterior. Em outras palavras, se Bn = An = "1"

será gerado um CARRY neste estágio, independente do que vier do estágio anterior. Esta condição de ser gerado um CARRY será denominada Gn e temos:

$$Gn = Bn \cdot An$$

b) se Bn = "1" e An = "Ø", ou Bn = "Ø" e An = "1", o CARRY C_{n-1} que vem do estágio anterior propagar-se-á através deste estágio. Isto quer dizer que Cn será "1" se C_{n-1} for "1" e "Ø" se C_{n-1} for "Ø". Esta condição de propagação do CARRY será denominada Pn e temos:

$$Pn = Bn \oplus An$$

c) considerando o exposto acima, podemos afirmar que o estágio n irá gerar um CARRY se Gn for "1" ou se Pn for "1" "E" houver um CARRY C_{n-1} vindo do estágio anterior, ou seja

$$Cn = Gn + Pn \cdot C_{n-1}$$

Vamos agora considerar a situação para cada estágio em particular. As equações abaixo exprimem os resultados para cada estágio, no que diz respeito ao CARRY.

$$C_0 = G_0 + P_0 \cdot C_{-1} \text{ (no caso } C_{-1} = \text{"Ø"})$$

$$C_1 = G_1 + P_1 \cdot C_0 = G_1 + P_1 G_0$$

$$C_2 = G_2 + P_2 \cdot C_1 = G_2 + P_2G_1 + P_2P_1G_0$$

e genericamente

$$Cn = Gn+PnG_{n-1} + PnP_{n-1}G_{n-2}+\ldots+PnP_{n-1}\ldots P_2P_1G_0$$

Como os termos Gn, Pn, G_{n-1}, P_{n-1}, ...,P_1, G_0 dependem somente das entradas do somador, a equação acima nos diz que é possível anteciparmos o valor do CARRY, considerando apenas as entradas do somador, sem necessidade de esperarmos que ele se propague por todos os estágios anteriores. Esta técnica de antecipação do CARRY é conhecida como CARRY LOOK-AHEAD e é utilizada para aumentar a velocidade do somador. Há circuitos integrados dedicados à implementação do CARRY LOOK-AHEAD, que podem ser utilizados em conjunção com os somadores.

Comparando a equação para Cn mostrada na figura 10.3:

$$Cn = Bn.An + C_{n-1} (Bn + An)$$

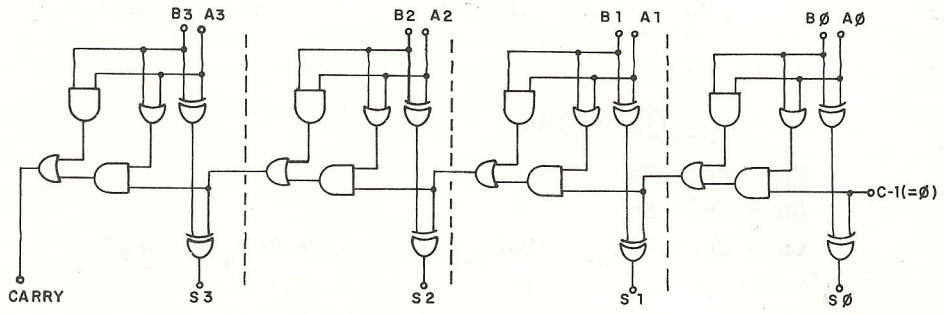
com aquela envolvendo os termos Gn e Pn

$$Cn = Gn + C_{n-1} \cdot Pn$$

vemos que Pn pode ser expresso também como a função OU das variáveis Bn e An, o que simplifica a construção do circuito.

$$Pn = Bn + An$$

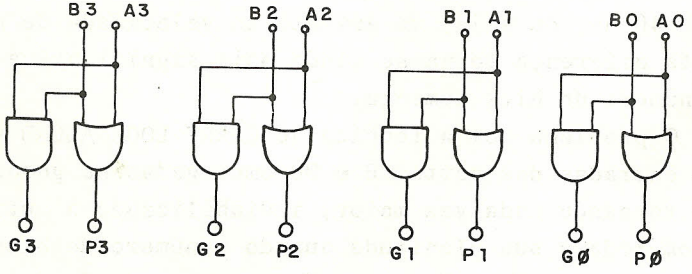
A figura 10.5 ilustra dois somadores de 4 bits, o primeiro com RIPPLE CARRY e o segundo com o CARRY LOOK AHEAD:

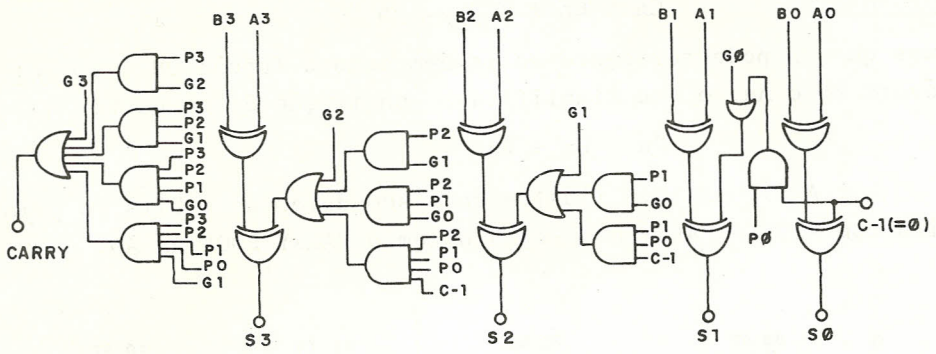


$$Sn = C_{n-1} \oplus Bn \oplus An$$

$$Cn = Bn \cdot An + C_{n-1} \cdot (Bn + An)$$

(a) Somador de 4 bits com RIPPLE CARRY





$$Sn = C_{n-1} \oplus Bn \oplus An$$

$$Pn = An + Bn$$

$$Gn = An \cdot Bn$$

$$Cn = Gn + PnG_{n-1} + PnP_{n-1}G_{n-2} + \ldots + PnP_{n-1}\ldots P_0C_{-1}$$

(b) Somador de 4 bits com CARRY LOOK-AHEAD

Fig. 10.5 - Somadores de 4 bits com RIPPLE CARRY e CARRY LOOK-AHEAD.

Admitindo um atraso de 10 ns por porta lógica, o somador com RIPPLE CARRY necessitaria de 90 ns para que a soma se completasse, ao passo que o somador com CARRY LOOK-AHEAD exigiria apenas 40 ns, ou seja, um aumento na velocidade de mais de 100%. Esta diferença torna-se ainda mais significativa à medida que o número de bits aumenta.

O problema com a técnica de CARRY LOOK-AHEAD é que o número de entradas das portas E e OU empregadas na geração dos Cn vai-se tornando cada vez maior, inviabilizando a utilização da mesma em toda a sua plenitude quando o número de bits a ser somados é muito grande. O que se faz usualmente é adotar uma solução intermediária utiliza-se a técnica de LOOK-AHEAD para grupos de bits e a de RIPPLE-CARRY entre os grupos.

10.3 - Subtração de Números Binários

A subtração de números binários é geralmente feita através da soma do complemento do subtraendo ao minuendo.

Vejamos o que vem a ser o complemento de um número binário. Define-se o complemento 2 de um número binário de n bits como:

$$\text{complemento } 2 = 2^n - \text{número}$$

Por exemplo, o complemento 2 de 1Ø1 é Ø11, o complemento 2 de 1Ø11 é Ø1Ø1, e assim por diante. Uma regra fácil para encontrar o complemento 2 de um número binário: troque os "Ø"'s por "1'"s e os "1'"s por "Ø"'s e some "1" ao resultado.

Define-se o complemento 1 de um número binário de n bits como:

$$\text{complemento } 1 = (2^n - 1) - \text{número}$$

Por exemplo, o complemento 1 de 1Ø1 e Ø1Ø, o complemento 2 de 1Ø11 e Ø1ØØ, e assim por diante. Para encontrar o complemento 1 de um número binário basta trocar os "Ø"'s por "1'"s e os "1'"s por "Ø"'s.

Voltando ao problema da subtração, suponhamos que se deseje efetuar a operação.

$$\begin{array}{rl} 51 = & 11ØØ11 \\ -\ 18 = & Ø1ØØ1Ø \end{array}$$

Para tanto somamos 51 ao complemento 1 de 18 e o CARRY resultante é novamente adicionado ao resultado. Este CARRY chama-se END-AROUND CARRY e a operação está indicada abaixo.

$$\begin{array}{lrl} & 51 & 11\ ØØ11 \\ \text{complemento 1 de} & 18 & \underline{1Ø\ 11Ø1} \quad + \\ & & 1Ø\ ØØØØ \\ \text{EAC} & & \underline{\qquad\quad 1} \\ & & 1Ø\ ØØØ1 = 33 \end{array}$$

A subtração também pode ser feita somando a 51 o complemento 2 de 18, conforme indicado.

	51	11	ØØ11
complemento 2 de	18	1Ø	111Ø
desprezado →	1	1Ø	ØØØ1 = 33

No caso da subtração ser feita através da soma do complemento 2 o EAC é desprezado.

Com base nestas idéias, podemos utilizar os somadores vistos anteriormente para realizar também subtrações. A figura 10.6 mostra um somador/subtrator para 4 bits, que realiza a subtração através da soma com o complemento 1 do subtraendo:

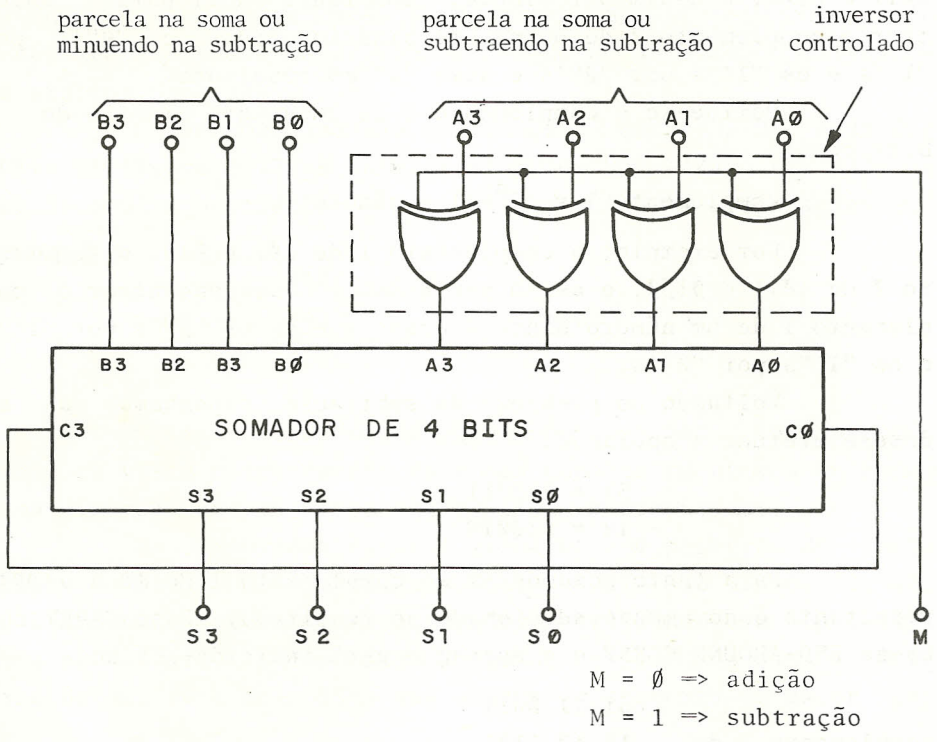


Fig.10.6 - Somador/subtrator para 4 bits operando com o complemento 1 para subtração.

No circuito da figura 10.6 a saída S será:

S = B + A para M = "Ø"
S = B - A para M = "1"

Observe a utilização das portas OU-EXCLUSIVO como inversores controlados. Esta é uma aplicação muito importante da função OU-EXCLUSIVO. Se a linha de controle M está em "Ø" a entrada An aparece sem inversão na saída da porta; se M está em "1", An é invertido. Este arranjo equivale a gerar o complemento 1 de A para a subtração. Observe também como o CARRY do último estágio é realimentado para a entrada de CARRY do primeiro estágio; isto equivale a soma do EAC ao resultado.

A figura 10.7 mostra o diagrama de um somador/subtrator para quatro bits que realiza a subtração através da soma do minuendo com o complemento 2 do subtraendo:

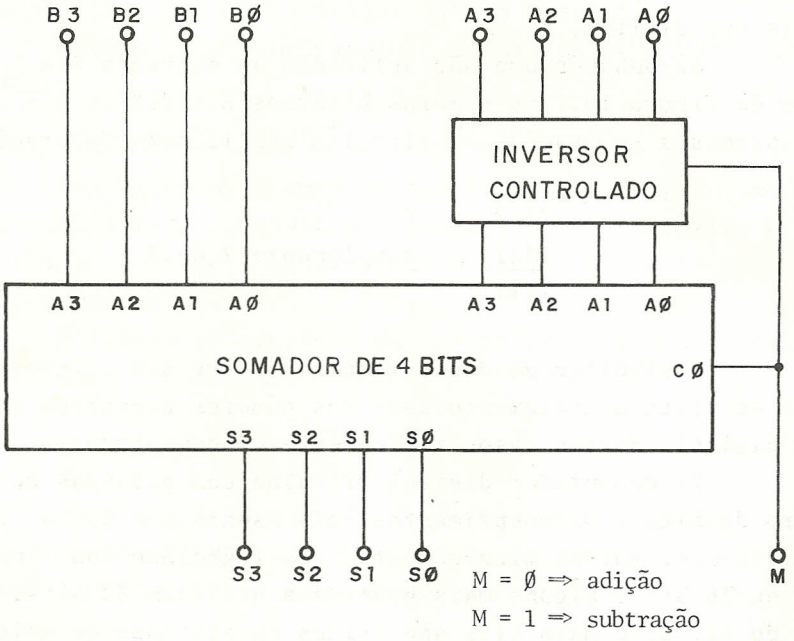


Fig. 10.7 - Somador/subtrator para 4 bits operando com o complemento 2 para a subtração.

Na figura 10.7 observamos que, se M = "1", a entrada A será invertida antes de ser aplicada ao somador; além disso, será somado "1" ao resultado. Este arranjo equivale a gerar o

complemento 2 de A para a subtração.

O circuito da figura 10.6 apresenta uma desvantagem em relação ao da figura 10.7. É que no primeiro devemos esperar que o EAC seja somado ao resultado, fazendo com que um tempo maior seja necessário para termos o resultado final disponível.

10.4 - Representação de Números Binários em Sistemas Digitais

No item anterior, verificamos o processo de subtração de números binários através da soma do complemento. Vamos examinar agora o problema da representação de números negativos em um sistema digital.

Suponhamos que são aplicadas às entradas B e A do circuito da figura 10.7 os números binários B = Ø1ØØ e A = Ø1Ø1 e que façamos a subtração. O circuito realizaria a operação:

```
   Ø1ØØ      B
+  1Ø11      complemento 2 de A
   1111
```

O significado deste resultado deve ser interpretado tendo em vista o tratamento dado aos números negativos em um sistema digital, nestes casos tipicamente um computador.

Um computador digital trabalha com palavras de certo número de bits. Os comprimentos mais usuais são 8, 16, 32, 48, 60 bits, etc. Muitos microcomputadores trabalham com palavras de 8 ou 16 bits; alguns mais poderosos utilizam 32 bits. Palavras de 48, 60 e mais bits são usadas em sistemas de maior porte.

Em qualquer caso, um bit, geralmente o bit mais significativo, é utilizado para indicar o sinal do número, é o chamado bit de sinal. A convenção usual é considerar-se o número positivo se o bit de sinal for igual a "Ø" e negativo se o bit de sinal for igual a "1". Com 8 bits, por exemplo, separando um bit para o sinal, poderíamos representar os números inteiros entre -127 e +127. Observe que isto não significa absolutamente

que um microcomputador com palavra de 8 bits esteja limitado a trabalhar com valores nesta faixa, uma vez que um número pode ser constituído de tantas palavras quantas se queira.

Os bits restantes, excluído o bit de sinal, representam a grandeza do número. Aqui também podemos escolher diversos modos de representação. São eles:

1. Sinal/magnitude - os $n-1$ bits, exceto o de sinal, representam o valor absoluto do número;

2. Complemento 1 - os $n-1$ bits, exceto o bit de sinal, representam
   - o valor absoluto do número, se este é positivo,
   - o complemento 1 do valor absoluto do número, se este é negativo;

3. Complemento 2 - os $n-1$ bits, exceto o bit de sinal, representam
   - o valor absoluto do número, se este é positivo,
   - o complemento 2 do valor absoluto do número, se este é negativo.

As três convenções são melhor esclarecidas através da tabela apresentada na figura 10.8, que mostra a interpretação de um número binário de 4 bits segundo as três convenções:

NÚMERO	INTERPRETAÇÃO		
BINÁRIO	SINAL/MAGNITUDE	COMPLEMENTO 1	COMPLEMENTO 2
0000	+ 0	+ 0	+ Ø
0001	+ 1	+ 1	+ 1
0010	+ 2	+ 2	+ 2
0011	+ 3	+ 3	+ 3
0100	+ 4	+ 4	+ 4
0101	+ 5	+ 5	+ 5
0110	+ 6	+ 6	+ 6
0111	+ 7	+ 7	+ 7
1000	- Ø	- 7	- Ø
1001	- 1	- 6	- 7
1010	- 2	- 5	- 6
1011	- 3	- 4	- 5
1100	- 4	- 3	- 4
1101	- 5	- 2	- 3
1110	- 6	- 1	- 2
1111	- 7	- 0	- 1

Fig. 10.8 - Interpretação de números binários em um computador digital.

A realização de operações aritméticas com números binários na forma sinal/magnitude é relativamente complexa e não nos deteremos neste ponto.

Observando o resultado da operação B - A proposta anteriormente, verificamos que este resultado 1111 (=-1) está perfeito, considerando que os números A e B estejam na forma de complemento 2. De fato, o circuito da figura 10.7 está capacitado a fornecer resultados corretos para quaisquer entradas A e B na forma de complemento 2, desde que é claro, o resultado esteja entre -7 e +7. Do mesmo modo, o circuito da figura 10.6 está capacitado a fornecer resultados corretos para quaisquer entradas A e B na forma de complemento 1, desde que se obedeça a restrição do resultado entre -7 e +7.

Tanto o circuito da figura 10.6 como o da figura 10.7 podem ser ampliados para maior número de bits, constituindo um somador/subtrator com maior capacidade. Por exemplo, se estendêssemos o circuito para 32 bits poderíamos trabalhar com números entre $+(2^{31} - 1)$ e $-(2^{31} - 1)$, ou seja, entre +2.147.483.649 e -2.147.483.649.

É interessante notar que o bit de sinal é parte integrante do número e como tal é também aplicado ao somador.

10.5 - Geração e Verificação de Paridade

É muito comum nos sistemas digitais o acréscimo de um bit de paridade a uma informação binária, para permitir a detecção de uma alteração que a informação possa sofrer quando é enviada de um ponto a outro.

Dizemos que uma informação binária possui paridade par quando o número de bits iguais a "1" for par. Se o número de bits iguais a "1" for ímpar, dizemos que a informação tem paridade ímpar.

O bit de paridade tem a finalidade de tornar a paridade das informações transmitidas entre as diversas partes do sistemas sempre par (ou sempre ímpar) de modo a permitir que ao ser recebidas em um determinado ponto sua integridade possa ser verificada.

Em geral, distinguimos na informação os bits de dados e o bit (ou bits) de paridade. Vamos supor que o dado 1Ø1ØØ11Ø deva ser transmitido através de uma linha de dados. Acrescentando um bit de paridade ímpar, transmitiríamos a informação.

P	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	Ø	1	Ø	Ø	1	1	Ø

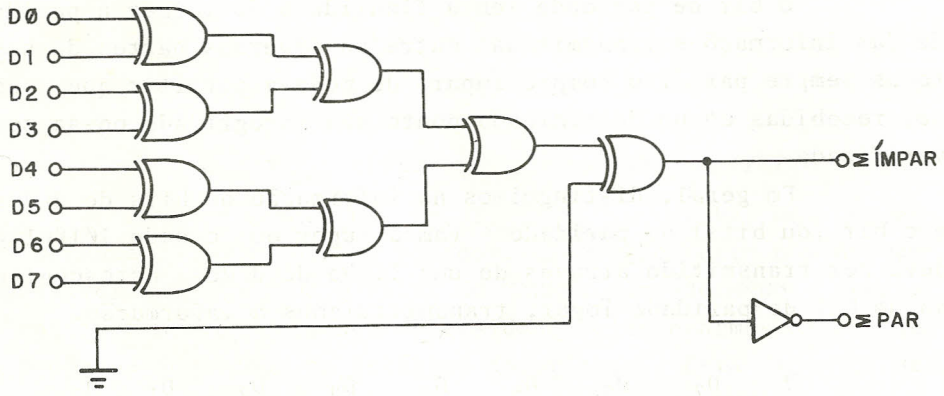
Quando a informação fosse recebida, o circuito que a recebesse calcularia novamente a paridade, incluindo no cálculo o bit de paridade. Se o resultado for paridade par, significa que houve um erro na transmissão.

O acréscimo de um bit de paridade é suficiente para permitir a deteção de erros de um bit; se dois bits se alterarem não será detetado o erro. No entanto, geralmente o acréscimo de apenas um bit de paridade assegura grande confiança na integridade da informação.

Em termos rigorosos, a paridade é o resultado da adição módulo 2 dos bits de dados. As regras da adição módulo 2 são muito simples:

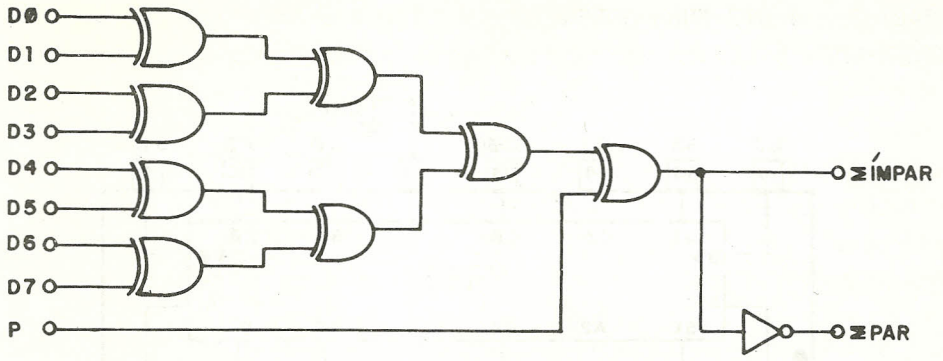
$$(0 + 0) \bmod 2 = 0$$
$$(0 + 1) \bmod 2 = 1$$
$$(1 + 0) \bmod 2 = 1$$
$$(1 + 1) \bmod 2 = 0$$

Portanto, a paridade pode ser gerada e verificada através de portas OU-EXCLUSIVO, conforme ilustram os circuitos da figura 10.9, para 8 bits de dados. O mesmo circuito é utilizado para geração e verificação da paridade, conforme ilustra a figura:



Desejando paridade ímpar, a saída Σ PAR é transmitida como bit de paridade; desejando paridade par, a saída ΣIMPAR é transmitida como bit de paridade.

(a) Geração do bit de paridade



Se a informação foi transmitida com paridade ímpar, a saída Σ PAR ativa indica erro; se a informação foi transmitida com paridade par, a saída Σ IMPAR ativa indica erro

(b) Verificação da paridade

Fig. 10.9 - Geração e verificação da paridade

O acréscimo de um bit de paridade permite detetar erros de um bit, mas não permite determinar o bit em erro. É possível adicionar mais bits de paridade, calculados sobre certos conjuntos de bits de dados e distribuídos convenientemente através da informação, de modo que seja possível detetar e corrigir erros de um único bit e detetar erros de mais de um bit. São os chamados códigos HAMMING, freqüentemente utilizados em sistemas de grande porte. O assunto é complexo e não será abordado aqui.

10.6 - Elementos Aritméticos da Família TTL

A série 74TTL oferece diversos circuitos integrados dedicados a funções aritméticas dos quais alguns estão descritos a seguir.

7483 - 4-BIT FULL ADDER

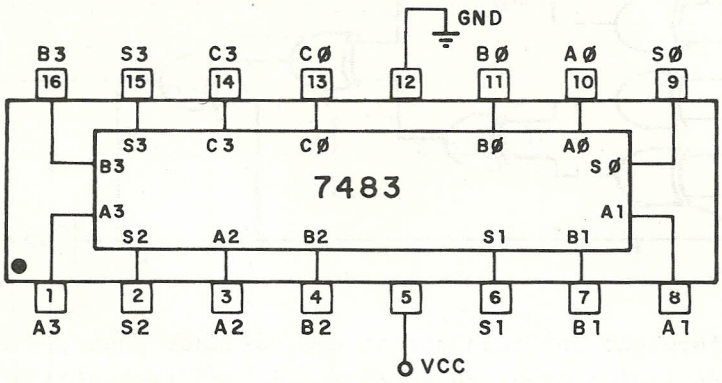


Fig. 10.10 - 7483.

O 7483 é um somador binário de 4 bits. C_0 é a entrada de CARRY do primeiro estágio e C_3 a saída de CARRY do último estágio. Diversos 7483 podem ser conectados em cascata, permitindo a construção de somadores para qualquer número de bits. Neste tipo de ligação $C_\emptyset$ de um CI recebe o C_3 do estágio anterior, quando se utiliza a técnica de RIPPLE CARRY.

Uma adição é completada em 16 ns, tipicamente, para cada circuito integrado.

7485 - 4-BIT MAGNITUDE COMPARATOR

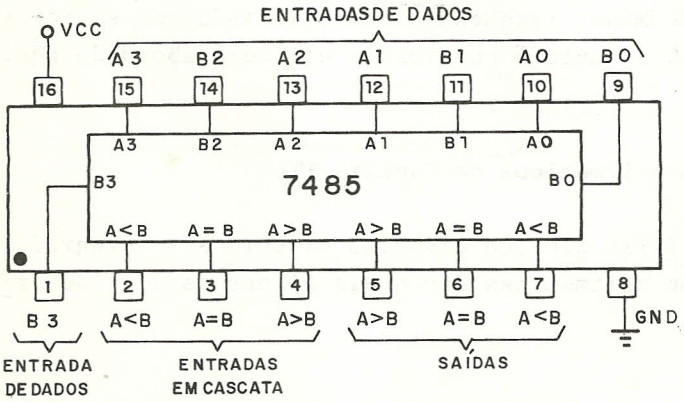


Fig. 10.11 - 7485.

Este CI é um comparador de quatro bits; ele compara as duas palavras aplicadas em suas entradas de dados e fornece a indicação de sua grandeza relativa. Diversos CI's podem ser conectados em cascata, possibilitando a comparação de palavras com qualquer número de bits. Neste caso, as saídas de comparação de um estágio são conectadas às entradas em cascata correspondentes do estágio seguinte. O resultado da comparação aparece nas saídas do último estágio. As entradas em cascata do primeiro estágio devem ser ligadas da seguinte maneira: (A=B)="1", (A<B)="Ø", (A>B)="Ø".

Se as duas palavras aplicadas ao CI são iguais, a saída A = B fica em "1" e as duas outras em zero; se a palavra em A é maior que a palavra em B, a saída A>B fica em "1" e as outras duas em zero e, finalmente, se a palavra em A é menor que a palavra em B, a saída A < B fica em "1" e as outras duas em "Ø".

O atraso de propagação típico é de 23 ns.

74180 - PARITY GENERATOR/TESTER

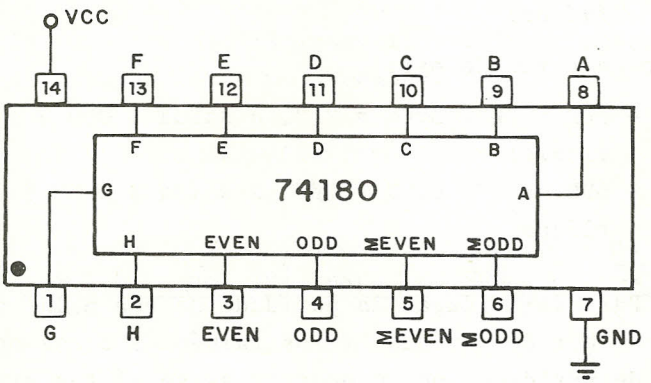


Fig. 10.12 - 74180.

Este circuito integrado é um gerador de paridade. As entradas A, B, C...,H são entradas de dados; as entradas EVEN e ODD controlam o funcionamento do circuito de acordo com a tabela apresentada na figura 10.13:

EVEN	ODD	Nº DE BITS EM 1 (entradas A,B,...,H)	Σ EVEN	Σ ODD
1	1	IRRELEVANTE	Ø	Ø
Ø	Ø	IRRELEVANTE	1	1
Ø	1	PAR	Ø	1
Ø	1	ÍMPAR	1	Ø
1	Ø	PAR	1	Ø
1	Ø	ÍMPAR	Ø	1

Fig. 10.13. - Funcionamento do 74180.

Para utilizar este circuito como gerador de paridade para uma palavra com 8 bits de dados temos diversas opções:

a) Geração de paridade ímpar:

- ODD = "1", EVEN = "Ø", a saída Σ ODD é o bit de paridade;
- ODD = "Ø", EVEN = "1", a saída Σ EVEN é o bit de paridade;

b) Geração de paridade par:

- ODD = "Ø", EVEN = "1", a saída Σ ODD é o bit de paridade;
- ODD = "1", EVEN = "Ø", a saída Σ EVEN é o bit de paridade.

Para verificação da paridade de uma palavra com 8 bits de dados e 1 bit de paridade temos também diversas opções. Se P é o bit de paridade, podem ocorrer as seguintes situações:

a) a paridade deve ser ímpar

- EVEN = P, ODD = $\overline{P}$, a saída Σ ODD = "1" indica erro de paridade;
- EVEN = $\overline{P}$, ODD = P, a saída Σ EVEN = "1" indica erro de paridade;

b) a paridade deve ser par

- EVEN = P, ODD = $\overline{P}$, a saída Σ EVEN = "1" indica erro de paridade;
- EVEN = $\overline{P}$, ODD = P, a saída Σ ODD = "1" indica erro de paridade.

Caso tenhamos menos de 8 bits de dados as entradas não utilizadas devem ser aterradas. É claro que isto deve ser feito tanto do lado da geração como da verificação igualmente, exceto que na verificação o bit de paridade pode ser diretamente aplicado em uma entrada. Podemos ligar entradas não usadas em "1", mas, neste caso, é preciso verificar se a paridade não é afetada.

O atraso de propagação é de 45 ns.

74181 - ARITHMETIC LOGIC UNIT

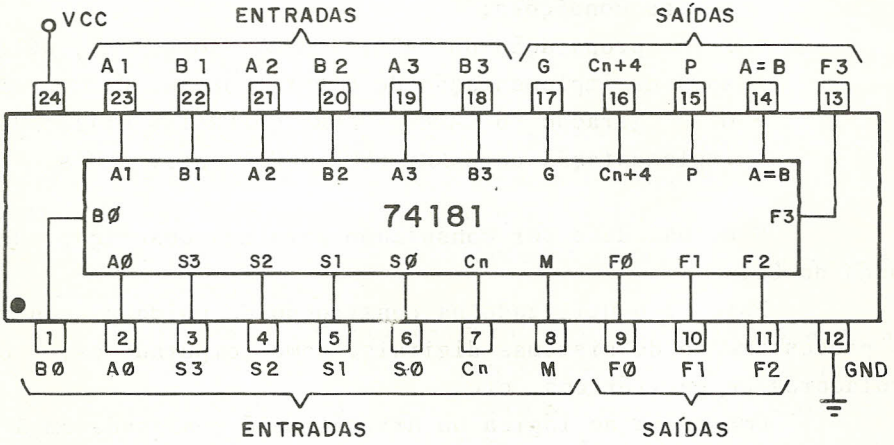


Fig. 10.14 - 74181.

Este circuito integrado é uma unidade aritmética e lógica (ALU) de quatro bits capaz de realizar diversas operações lógicas e aritméticas com os operandos presentes nas entradas A e B. Tantas ALU'S quantas necessárias podem ser ligadas em cascata, permitindo trabalhar com palavras de qualquer número de bits.

São os seguintes os sinais de entrada para a ALU:

- B_3, B_2, B_1, $B_\emptyset$ - operando B;
- A_3, A_2, A_1, $A_\emptyset$ - operando A;
- S_3, S_2, S_1, $S_\emptyset$ - seleciona a função lógica ou aritmética a ser realizada pela ALU;
- M - seleciona o modo de operação da ALU;
- M = "Ø" - operações aritméticas;
- M = "1" - operações lógicas;
- Cn - entrada de CARRY da ALU (CARRY-IN).

São os seguintes os sinais de saída da ALU:

- F_3, F_2, F_1, $F_\emptyset$ - resultado da operação lógica ou aritmética realizada com os operandos;
- C_{n+4} - saída de CARRY da ALU (CARRY-OUT);
- A = B - indica a igualdade dos operandos A e B em certas condições;
- P - propagação do CARRY (CARRY PROPAGATE), utilizada na implementação de CARRY LOOK-AHEAD entre ALU's
- G - geração do CARRY (CARRY GENERATE) utilizada na implementação de CARRY-LOOK-AHEAD entre ALU's.

O manual deve ser consultado para uma descrição detalhada do CI.

Este CI é utilizado na construção da unidade central de processamento de sistemas digitais, como computadores, controladores de periféricos, etc.

Uma operação lógica ou aritmética é realizada em 35 ns.

74182 - LOOK-AHEAD CARRY GENERATOR

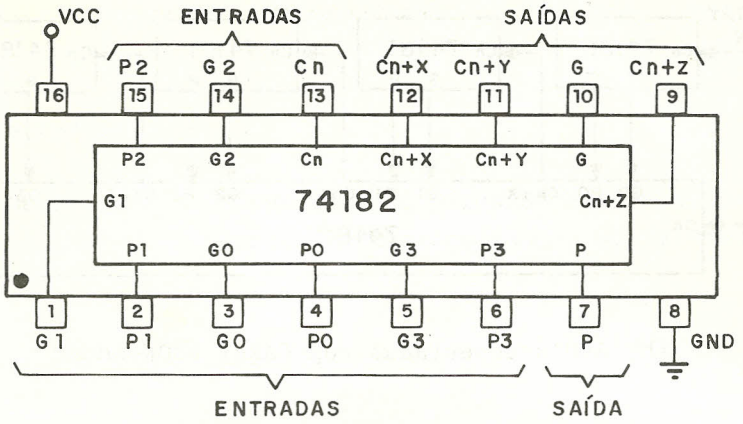


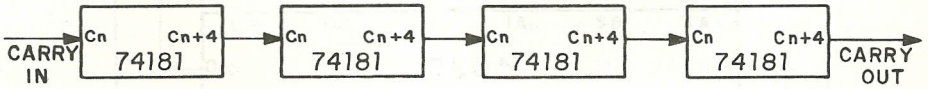
Fig. 10.15 - 74182.

Este circuito integrado é um gerador de CARRY LOOK-AHEAD utilizado para tornar mais rápida a operação de somadores.

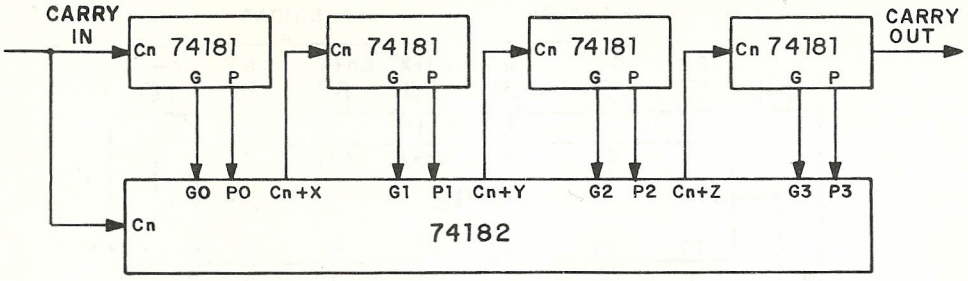
As entradas G_0, P_0, G_1, P_1, G_2, P_2, G_3 e P_3 são os termos de geração e propagação do CARRY para os diversos estágios. Cn é a entrada de CARRY.

As saídas G e P são destinadas a permitir a ligação de diversos CI's em cascata, quando necessário. C_{n+x}, C_{n+y} e C_{n+z} são os CARRIES para os estágios intermediários.

Com um CI 74182 é possível fazer um esquema de antecipação de CARRY para quatro ALU's 74181. A figura 10.16 ilustra esta aplicação e esclarece melhor a utilização deste circuito integrado:



(a) ALU's conectadas com RIPPLE CARRY



(b) ALU's conectadas com CARRY LOOK-AHEAD

Fig. 10.16 - Conexão do 74182.

O atraso deste circuito integrado é tipicamente de 13 ns.

10.7 - Elementos Aritméticos da Família CMOS

Também na série 4ØØØ CMOS encontramos elementos aritméticos, alguns dos quais são descritos a seguir.

4032 - TRIPLE SERIAL ADDER (POSITIVE LOGIC)

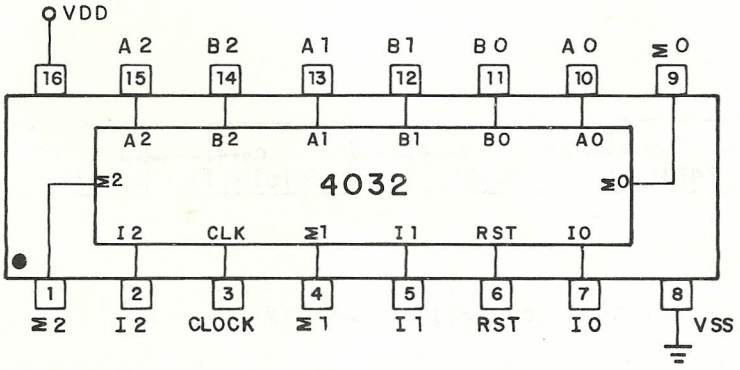


Fig. 10.17 - 4032.

Este circuito integrado contém três somadores e é capaz de adicionar palavras binárias de qualquer comprimento um bit por vez.

Para operação normal, as entradas de inversão I são mantidas em "Ø". A entrada RST deve ficar em "1" durante uma transição positiva do CLOCK, de modo a limpar o CARRY remanescente de uma operação anterior. Em seguida, a linha RST e o CLOCK devem voltar para "Ø".

As palavras a serem somadas devem ser apresentadas ao somador um bit por vez. A soma bit a bit de A e B aparece nas saídas Σ; um CLOCK é requerido para a soma de cada bit. As palavras em A e B devem ser apresentadas ao somador começando pelo LSB e terminando com o bit de sinal.

O tempo requerido para a soma de cada bit é de 250 ns para V_{DD} = 10V e 800 ns para V_{DD} = 5V.

4063 - 4-BIT MAGNITUDE COMPARATOR

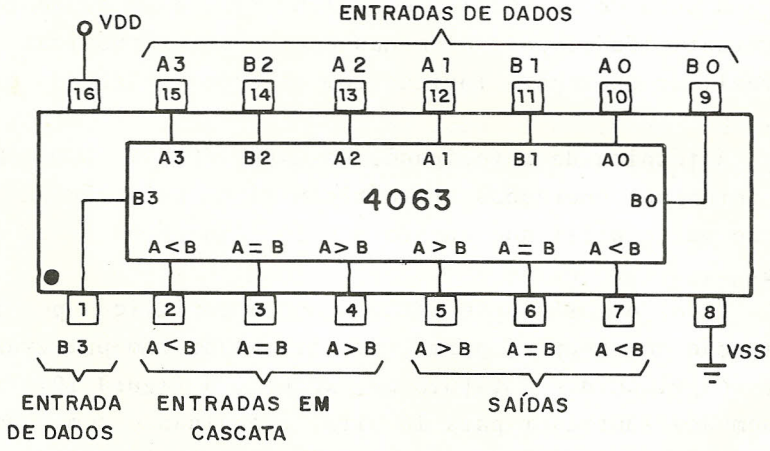


Fig. 10.18 - 4063.

O 4063 é um comparador de 4 bits; ele compara duas palavras aplicadas em suas entradas de dados e fornece a indica

ção de sua grandeza relativa.

Este CI é o equivalente CMOS do 7485. Valem as mesmas observações feitas na descrição deste último.

O tempo de comparação é de 250 ns para V_{DD} = 10V e 625 ns para V_{DD} = 5V.

10.8 - Aplicações

As aplicações dos elementos aritméticos são vastíssimas. Eles são extensivamente aplicados em computadores digitais, perfazendo toda sorte de funções relacionadas à manipulação de grandezas numéricas representadas em binário. É importante destacar que a multiplicação de números binários pode ser feita através de somas sucessivas; a divisão pode ser feita através de subtrações sucessivas. Assim, o somador é um elemento básico que permite a implementação de todas as operações aritméticas, desde que haja a correspondente lógica adicional de controle.

A unidade aritmética e lógica 74181 é um elemento de grande versatilidade, pois além das operações aritméticas pode também realizar operações lógicas com os dados colocados em sua entrada.

A técnica de antecipação do CARRY (CARRY LOOK-AHEAD) permite tornar as operações aritméticas ainda mais rápidas e é um recurso muito utilizado quando a velocidade é um fator importante.

Todos os CI's apresentados neste capítulo possibilitam a ligação em cascata, permitindo trabalhar com palavras de qualquer comprimento. A título de exemplo, a figura 10.19 ilustra um somador/subtrador para 16 bits, trabalhando com números representados na notação de complemento 2, e implementado com circuitos integrados TTL. O circuito fará as operações B + A para M = "Ø" e B - A para M = "1". Os inversores controlados são portas OU-EXCLUSIVO 7486.

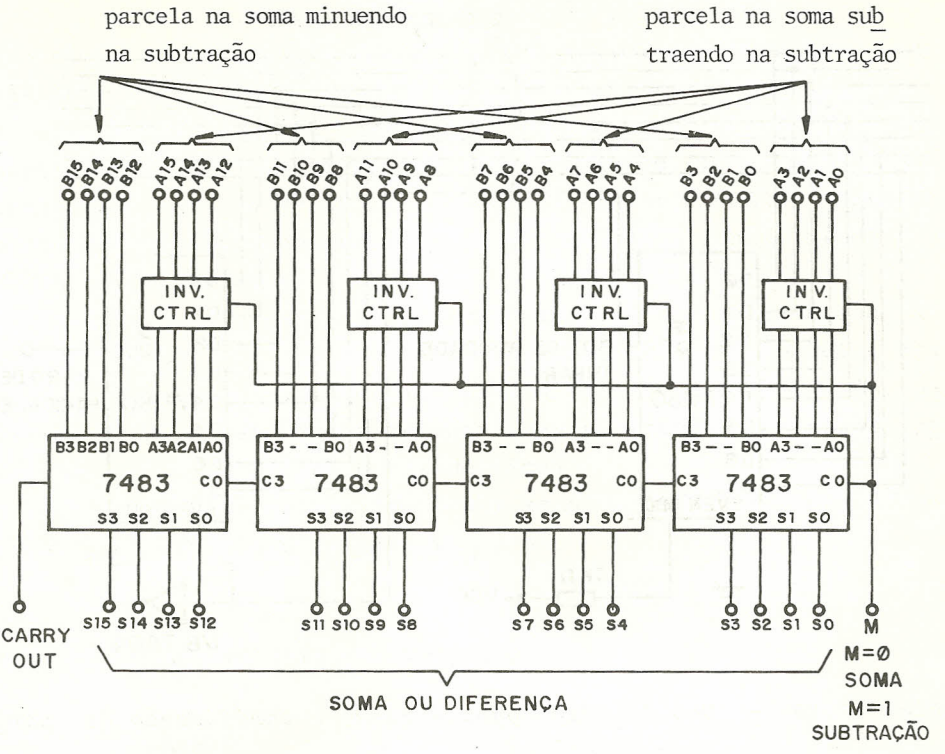


Fig. 10.19 - Somador/subtrator para 16 bits para números binários representados na forma de complemento 2.

O uso do circuito integrado 74180 para geração e verificação de paridade é ilustrado na figura 10.20. Nesta figura, mostra-se uma das maneiras de utilizá-lo para gerar e verificar a paridade ímpar de uma palavra binária de 8 bits.

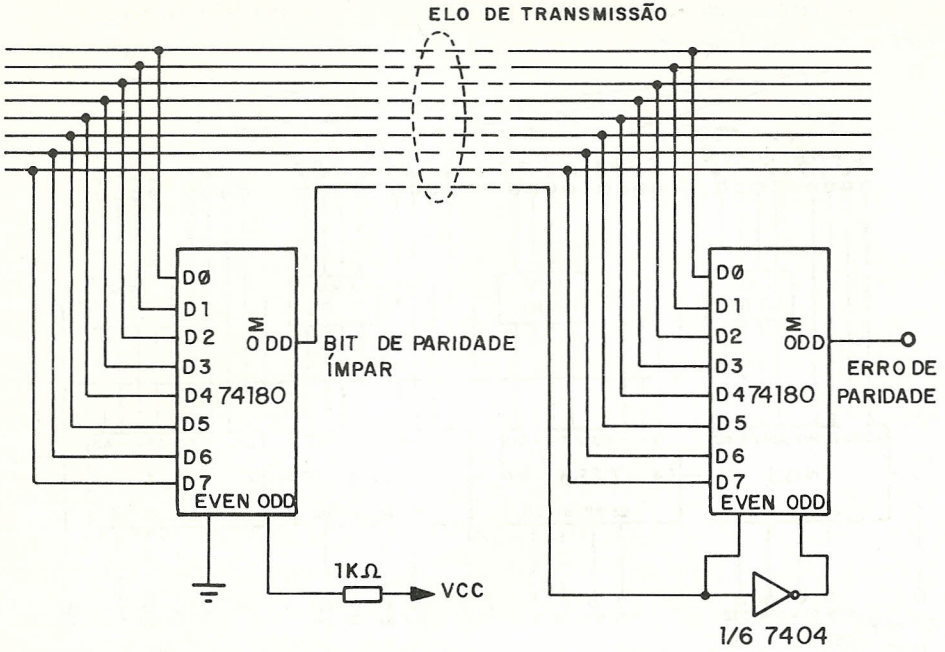


Fig. 10.20 - Uso do 74180 para geração e verificação de paridade ímpar.

CAPÍTULO 11 - MEMÓRIAS SEMICONDUTORAS

11.1 - Introdução

Memórias são dispositivos capazes de armazenar informações. A figura 11.1 ilustra o diagrama básico de uma memória:

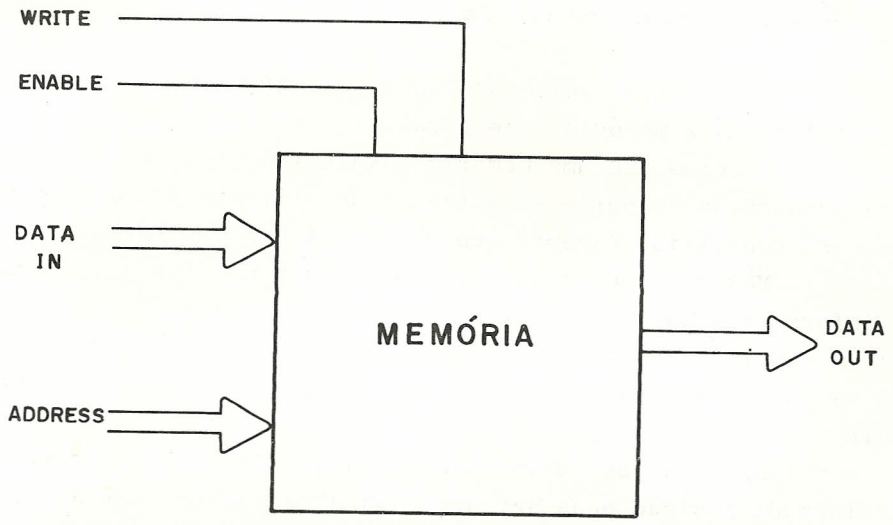


Fig. 11.1 - Diagrama básico de uma memória.

A memória contém diversas posições, cada uma das quais indicada por um endereço (ADDRESS); se a memória possui n linhas de endereço, conterá 2^n posições. Em cada uma das posições é possível armazenar um certo número de bits, geralmente 1, 4 ou 8.

Para que o conteúdo de uma determinada posição seja lida, a memória é habilitada através da ativação da linha ENABLE e o endereço desejado é colocado nas linhas ADDRESS. Após um determinado intervalo de tempo, o conteúdo da posição endereçada aparece nas linhas de saída DATA OUT.

Na operação de escrita, a memória é habilitada, o endereço desejado é colocado nas linhas ADDRESS e a informação que

se quer armazenar é colocada nas linhas DATA IN. Em seguida, aplica-se um pulso na linha de controle WRITE, o que faz com que a informação seja armazenada na memória.

As memórias são parte essencial de qualquer sistema digital de maior complexidade. Neste capítulo, apresentaremos os diversos tipos de memórias semicondutoras, suas características e ilustraremos algumas aplicações.

11.2 - Características Gerais

Neste item, veremos algumas características e definições relativas a memórias em geral.

Dizemos que uma memória é volátil quando a informação nela armazenada se perde se a tensão de alimentação for removida; caso contrário, dizemos que a memória é não-volátil.

Memórias de acesso seqüencial são aquelas nas quais as informações são acessadas em seqüência; isto quer dizer que, para acessarmos uma determinada posição, devemos passar por todas as posições anteriores. Este caso seria ilustrado tipicamente por uma fita magnética.

Memórias de acesso aleatório são aquelas nas quais uma determinada posição pode ser acessada diretamente, sem passar pelas posições anteriores. Este é o caso das memórias semicondutoras que veremos nos itens seguintes. As memórias de acesso aleatório são inerentemente mais rápidas que as de acesso seqüencial.

O tempo de acesso de uma memória é definido como o intervalo de tempo decorrente entre a aplicação do endereço e a apresentação dos dados nas linhas de saída. Este tempo pode variar desde alguns nanosegundos até algumas centenas de nanosegundos para as memórias semicondutoras e pode atingir até vários segundos, para uma fita magnética, por exemplo.

A capacidade de uma memória é a quantidade de informações que ela pode armazenar. Em geral, a capacidade é expressa em termos de Kbits ou KBYTES, sendo interessante notar que o "K" significa 1024, e não 1000 como é usual. Por exemplo, uma memória de 2KBYTES é capaz de armazenar 2048 bytes. Atualmente

as memórias semicondutoras de maior capacidade são capazes de armazenar 64 Kbits (65536 bits), prevendo-se para breve a introdução de memórias de 256Kbits. Para discos e fitas magnéticas a capacidade pode atingir centenas de megabytes.

Dizemos que uma memória é dinâmica quando as informações devem ser reescritas periodicamente, sob pena de se perderem. Este é o caso das memórias semicondutoras de maior capacidade, onde um bit é armazenado sob forma de uma carga em um capacitor. Com o correr do tempo, esta carga vai se esvaindo e é necessário reescrever a informação periodicamente em cada endereço. Esta operação é conhecida como REFRESH e, tipicamente, deve ser executada a cada 2 ms. Por outro lado, as memórias que não necessitam de REFRESH periódico são denominadas estáticas. Nestas memórias a informação é geralmente armazenada em FLIP-FLOPS.

11.3 - Memórias ROM

As memórias ROM (READ-ONLY MEMORY) são memórias semicondutoras nas quais as informações podem ser apenas lidas; não se pode escrever informações em uma ROM. Evidentemente, as memórias ROM são não-voláteis e a informação está nelas armazenada de maneira permanente.

Embora possa parecer estranho a primeira vista a existência de uma memória de leitura apenas, as memórias ROM encontram enorme aplicação na prática. Para ilustrar este ponto, considere por exemplo o caso de um microcomputador. Quando o ligamos, é necessário que ao menos um programa inicial esteja disponível, permitindo que outros programas sejam carregados a partir de uma fita cassete ou disco flexível (FLOPPY DISK). Este programa inicial deve estar permanentemente na memória e nunca será alterado, logo a solução mais óbvia é armazená-lo em uma ROM.

De maneira ampla, podemos dizer que as memórias ROM são utilizadas para armazenar informações que nunca (ou muito raramente) serão alteradas e devem estar sempre disponíveis; estas informações podem ser programas, tabelas, etc.

Há vários tipos de memórias ROM:

a) MASK-ROM

Estas memórias são fabricadas já contendo a informação desejada. O nome MASK-ROM deriva do fato de que as máscaras utilizadas na fabricação do circuito integrado são projetadas pelo fabricante sob encomenda do usuário. O desenvolvimento das máscaras é muito caro e assim o uso das MASK-ROM's só é viável em aplicações de grande volume. As configurações mais comuns são 1K x 8 (1024 posições, 8 bits por posição), 2K x 8, 4K x 8 e 8K x 8. O tempo de acesso é, em geral, de 450 ns a 700 ns.

b) PROM (PROGRAMMABLE READ-ONLY MEMORY)

As memórias PROM são programáveis pelo usuário e após programadas não podem ser alteradas. Inicialmente, todos os bits estão em "1", ou em "Ø". Nas memórias PROM cada bit corresponde a um minúsculo fusível e a programação consiste em romper os fusíveis correspondentes aos bits que se deseja levar ao estado contrário ao inicial. Como os fusíveis podem ser rompidos mas não recompostos, a programação só pode ser feita uma única vez.

Mesmo quando duas PROM's são eletricamente compatíveis e possuem a mesma disposição de pinos (PIN OUT), a maneira de programá-las pode ser diferente, se elas forem de fabricantes diferentes. Entretanto, de modo geral, podemos dizer que a programação de cada endereço é feita do seguinte modo: com a memória não habilitada, e o endereço que se quer programar colocado nas linhas de endereço, aplica-se um pulso de grande amplitude (20 a 30V) e fornece-se uma corrente elevada (20 a 50 mA) às saídas correspondentes aos bits que se deseja levar ao estado contrário ao inicial.

Encontra-se PROM's com capacidades desde 32 até 8192 palavras de 4 ou 8 bits e tempos de acesso desde 100 ns até 700 ns, tipicamente.

As PROM's são utilizadas quando a informação que se deseja armazenar já está perfeitamente definida e sabidamente correta, e o uso de MASK-ROM's é inviável. Durante o desenvol

vimento de um projeto, o uso de PROM's não é interessante, pois se após a programação verificar-se a necessidade de alguma alteração a única alternativa é substituir o circuito integrado e gravar um outro.

c) EPROM (ERASABLE PROGRAMMABLE READ-ONLY MEMORY)

As memórias EPROM são as mais populares das ROM's.Elas podem ser programadas pelo usuário e, se necessário,apagadas e novamente programadas, sendo possível repetir o processo um número ilimitado de vezes. Esta característica torna a utilização das memórias EPROM especialmente indicada durante o desenvolvimento de um projeto.

A programação de uma EPROM é feita de maneira similar a de uma PROM. O apagamento é conseguido expondo-se a EPROM à luz ultravioleta. Para permitir a incidência da luz no interior do circuito integrado, as EPROM's possuem uma janela de quartzo transparente; quando em uso, esta janela deve ser coberta por um papel opaco, evitando assim o apagamento acidental da memória, devido a exposição à luz solar ou fluorescente.

As configurações mais comuns das memórias EPROM são 256 x 8, 1K x 8, 2K x 8, 4K x 8 e 8K x 8 e os tempos de acesso variam de 350 ns a 650 ns, tipicamente. Vejamos alguns dados relativos às memórias EPROM mais comuns.

2708 - 8K (1K x 8) UV ERASABLE PROM

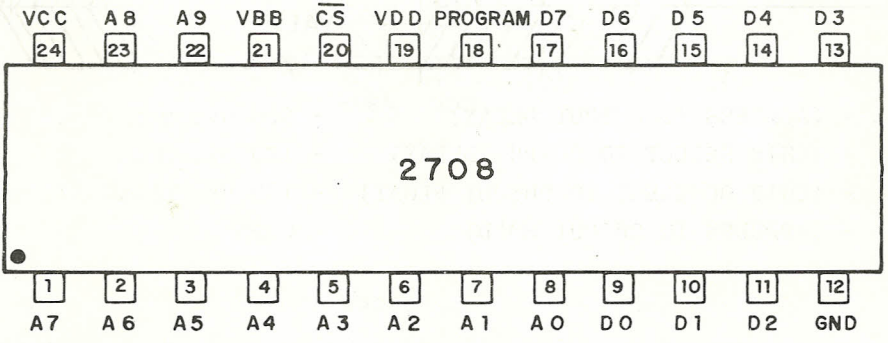
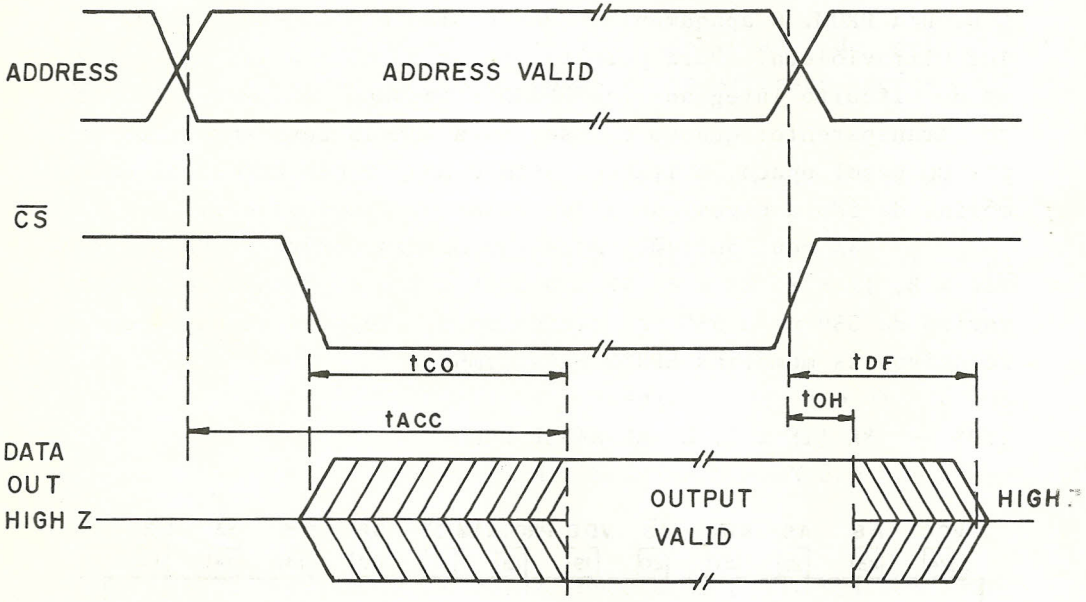


Fig. 11.2 - 2708.

Esta EPROM contém 1024 posições de 8 bits e requer três tensões de alimentação V_{CC} = +5V, V_{DD} = + 12V e V_{BB} = -5V. A_0 a A_9 são as linhas de endereçamento e D_0 a D_7 as linhas de saída; as saídas são TRIESTATE e $\overline{CS}$ é a linha de habilitação, ativa no estado lógico "Ø". O pino 18 somente é utilizado durante a programação e permanece em "Ø" durante a operação normal.

Há diversas versões da EPROM 2708 (2708, 2708-1, 2708-2, etc) com tempos de acesso variando entre 350 ns e 550 ns e dissipação de potência entre 425 mW e 800 mW. Em todos os casos, as entradas e saídas são compatíveis com TTL.

A figura 11.3 ilustra os diagramas de tempo para a EPROM 2708.



t_{ACC} - (ADDRESS TO OUTPUT DELAY) - 450 ns, max.
t_{CO} - (CHIP SELECT TO OUTPUT DELAY) - 120 ns, max.
t_{DF} - (CHIP DESELECT TO OUTPUT FLOAT) - 120 ns, max.
t_{OH} - (ADDRESS TO OUTPUT HOLD) - 0 ns

Fig. 11.3 - Diagrama de tempo para a EPROM 2708.

Este diagrama nos mostra que, após a aplicação do endereço desejado, as informações estarão disponíveis na saída após um tempo t_{ACC} de no máximo 450 ns. É claro que a memória deve estar selecionada, para que as saídas deixem o estado de alta impedância, e entre o selecionamento da memória e as saídas estarem válidas decorre um intervalo de tempo t_{CO} de, no máximo 120 ns. Quando a linha $\overline{CS}$ é desativada, decorre um tempo t_{DF} de no máximo 120 ns para que as saídas retornem ao estado de alta impedância. Quando as linhas de endereço se tornam inválidas, as linhas de saída mudam de estado imediatamente, o que é denotado pelo parâmetro $t_{OH} = \emptyset$ ns.

2716 - 16K (2K x 8) UV ERASABLE PROM

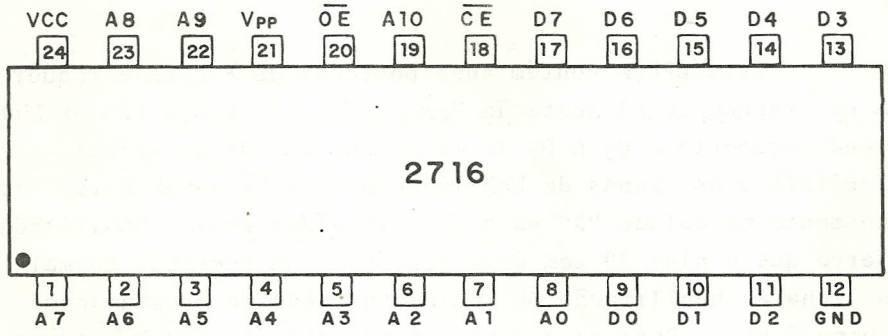


Fig. 11.4 - 2716.

Esta EPROM contém 2048 posições de 8 bits e requer apenas uma tensão de alimentação V_{CC} = +5V, A_0 a A_{10} são as linhas de endereçamento e D_0 a D_7 as linhas de saída; as saídas são TRI-ESTATE e as linhas $\overline{CE}$ e $\overline{OE}$ devem estar simultaneamente ativas no estado "Ø" para que as saídas estejam habilitadas. Vpp é utilizado somente para programação e no uso normal deve ser conectado a V_{CC}. Há diversas versões desta EPROM (2716, 2716-1, 2716-2, etc) com tempos de acesso variando entre 350 ns e 650 ns; a dissipação de potência é de no máximo 525 mW com a memória selecionada e 132 mW com a memória não selecionada. Em todos os casos, as entradas e saídas são compatíveis com TTL.

2732 - 32K (4K x 8) UV ERASABLE PROM

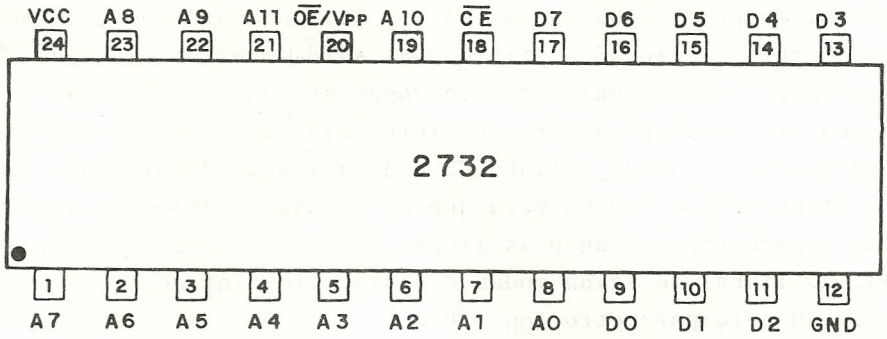


Fig. 11.5 - 2732.

Esta EPROM contém 4096 posições de 8 bits e requer apenas uma tensão de alimentação V_{CC} = +5V. A_0 a A_{11} são as linhas de endereçamento e D_0 a D_7 as linhas de saída; as saídas são TRI-ESTATE e as linhas de habilitação $\overline{CE}$ e $\overline{OE}$ devem estar simultaneamente no estado "Ø" para que as saídas sejam habilitadas. Observe que o pino 20 tem duas funções. Em operação normal é uma linha de habilitação $\overline{OE}$; quando no modo de programação, este pino é V_{PP}. Para as diversas versões desta EPROM o tempo de acesso varia entre 200 ns e 550 ns; a dissipação de potência é de 750 mW com a memória ativa e 150 mW com a memória não selecionada. Em todos os casos, as entradas e saídas são compatíveis com TTL.

Observe que a disposição dos pinos nas EPROM's 2716 e 2732 é quase a mesma, diferindo apenas a função do pino 21 (V_{PP} na 2716 e A_{11} na 2732). Assim, é muito simples expandir a capacidade de armazenamento de um sistema que utilize 2716, substituindo-as por 2732 e modificando apenas as conexões de um pino.

d) EAROM (ELECTRICALLY ALTERABLE READ-ONLY MEMORY)

As memórias EAROM são programáveis pelo usuário e podem ser eletricamente apagadas e reprogramadas quantas vezes necessário. Estas memórias possuem um custo muito elevado e, assim, são utilizadas somente em aplicações nas quais a caracte

rística de reprogramação com a memória no circuito, sem necessidade de removê-la, seja importante. Um caso típico é o armazenamento dos parâmetros de operação de um terminal de vídeo (velocidade, endereço, etc). Estes parâmetros dependem da linha e do protocolo que se usa para comunicação com o terminal. Utilizando uma EAROM é possível alterá-los via teclado, sem necessidade de modificar conexões internas ao terminal, o que é muito prático.

11.4 - Memórias RAM

Ao contrário das memórias ROM, as memórias RAM (RANDOM ACCESS MEMORY) ou RWM (READ/ WRITE MEMORY) podem ser lidas e escritas. Estas memórias são utilizadas para armazenar informações (programas, dados, etc) que devem ser modificadas no correr do tempo.

O nome RAM não é o mais correto, embora seja o mais utilizado. Esta sigla refere-se ao tipo de acesso, que é aleatório ou randômico, e tanto as memórias RAM como as ROM se enquadram nesta categoria. Assim, a denominação mais apropriada para estas memórias seria RWM, salientando a característica de poderem ser lidas e escritas. Entretanto, como a denominação RAM é a mais difundida, será esta a que utilizaremos aqui.

As configurações mais comuns para as memórias RAM são 128, 256, 1024, 2048, 4096 ou 16384 palavras de 1, 4 ou 8 bits. Em geral, quando a capacidade é de 16 Kbits ou maior, a memória é dinâmica; se a capacidade é menor que 16 Kbits, a memória é estática.

A figura 11.6 ilustra o diagrama e a tabela funcional de uma RAM 2102, que é capaz de armazenar 1024 palavras de 1 bit:

$\overline{CE}$	R/$\overline{W}$	DIN	DOUT	MODE
1	X	X	HI-Z	DESELECTED
0	0	0	0	WRITE "Ø"
0	0	1	1	WRITE "1"
0	1	X	DOUT	READ

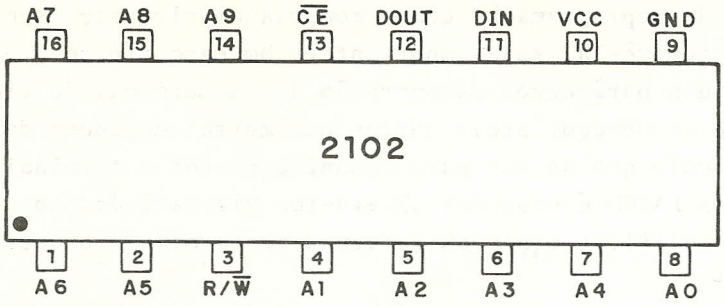
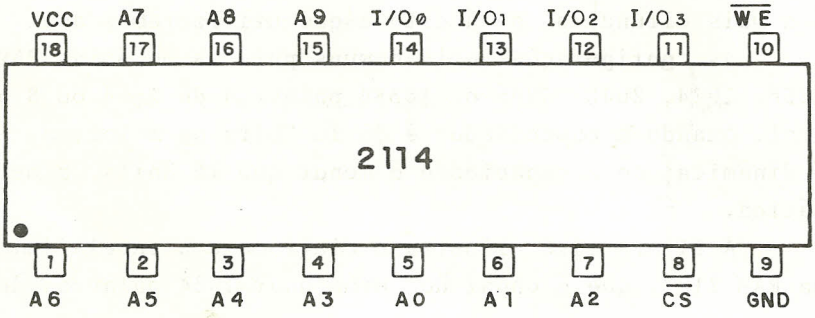


Fig. 11.6 - 2102 (1K X 1) e tabela funcional.

Esta memória RAM estática opera com uma tensão de alimentação de V_{CC} = +5V. As linhas A_0 a A_9 são as linhas de endereçamento, DIN a entrada de dados e DOUT a saída de dados. Esta saída é TRI-ESTATE e $\overline{CE}$ é a linha de habilitação, ativa no estado lógico "Ø". As operações de leitura e escrita são controladas pela linha R/$\overline{W}$; com a memória selecionada, se R/$\overline{W}$ = "1" ocorre uma operação de leitura e se R/$\overline{W}$ = "Ø" ocorre uma operação de escrita. Observe que durante a escrita, a memória está simultaneamente colocando dados na linha de saída.

Uma das memórias RAM estáticas mais populares é a 2114, cujo diagrama e tabela funcional estão ilustrados na figura 11.7:



$\overline{CS}$	$\overline{WE}$	I/O	MODE
1	X	HI-Z	DESELECTED
0	1	OUT	READ
0	0	IN	WRITE

Fig. 11.7 - RAM 2114 (1K X 4) e tabela funcional.

A RAM 2114 contém 1024 palavras de 4 bits e opera com uma tensão de alimentação V_{CC} = +5V. As linhas A_0 a A_9 são as linhas de endereço. As linhas I/O_0 a I/O_3 são bidirecionais; durante uma operação de escrita elas servem como entrada e durante uma operação de leitura servem como saída. Quando a linha de habilitação está inativa ($\overline{CS}$ = "1"), as linhas I/O estão no estado de alta impedância. A linha $\overline{WE}$ controla as operações de leitura e escrita, com $\overline{WE}$ = "Ø" tem-se uma operação de escrita e com $\overline{WE}$ = "1" tem-se uma operação de leitura.

Há diversas versões da memória 2114, com tempos de acesso variando entre 100 ns e 250 ns. A figura 11.8 ilustra o diagrama de tempos de uma 2114 para uma operação de leitura:

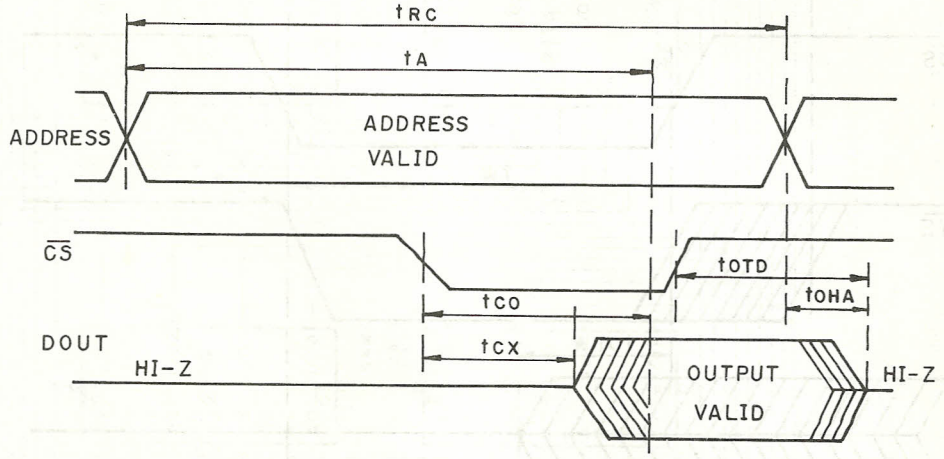


Fig. 11.8 - Diagrama de tempos para leitura de uma 2114.

Os parâmetros da figura 11.8 e seus valores típicos para uma das versões são os seguintes:

(2114AL - 3)

t_{RC} (READ CYCLE TIME) - 150 ns, mínimo;
t_A (ACCESS TIME) - 150 ns, máximo;
t_{CO} (CHIP SELECTION TO OUTPUT VALID) - 70 ns, máximo;

t_{CX} (CHIP SELECTION TO OUTPUT ACTIVE) - 10 ns, mínimo;
t_{OTD} (OUTPUT 3-STATE FROM DESELECTION) - 40 ns, máximo;
t_{OHA} (OUTPUT HOLD FROM ADDRESS CHANGE) - 15 ns, mínimo.

A figura 11.9 mostra o diagrama de tempos para uma operação de escrita:

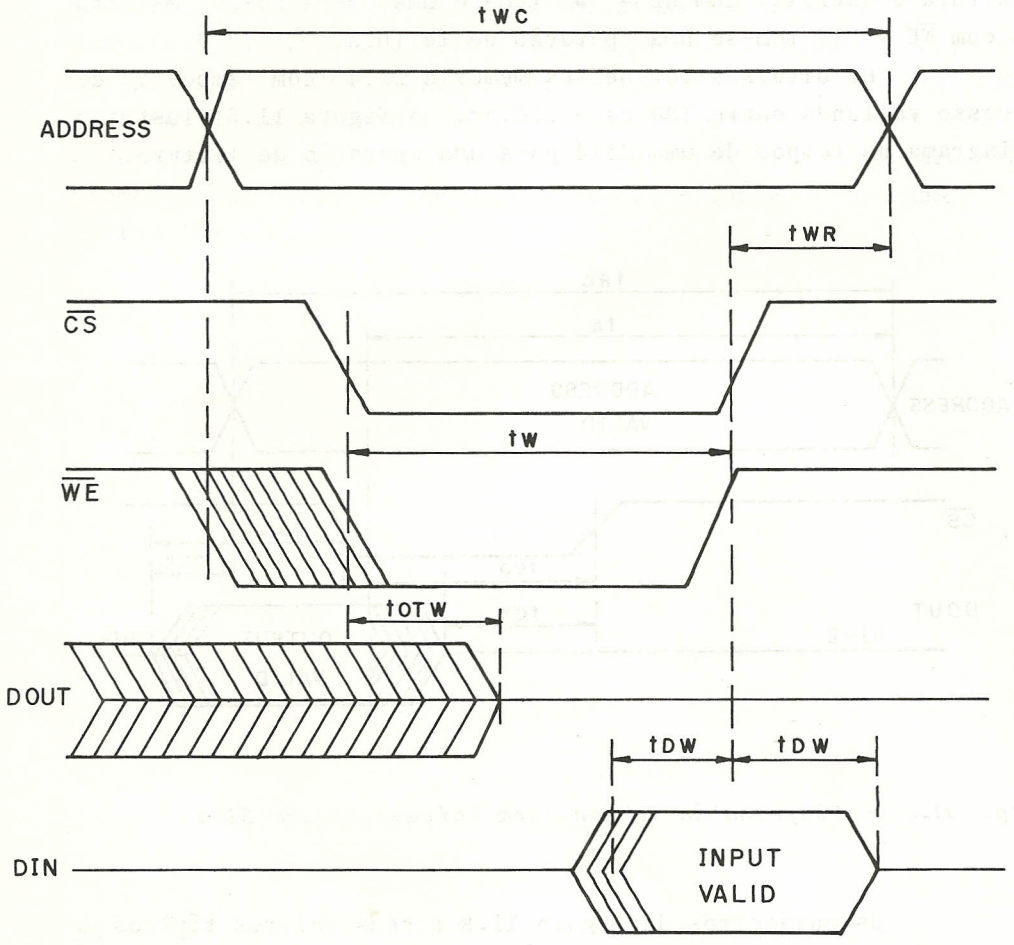


Fig. 11.9 - Diagramas de tempo para escrita de uma 2114.

Os parâmetros ilustrados na figura 11.9 e seus valores típicos para uma 2114 AL-3 são os seguintes:

t_{WC} (WRITE CYCLE TIME) - 150 ns, mínimo;
t_{W} (WRITE TIME) - 90 ns, mínimo;
t_{WR} (WRITE RELEASE TIME) - 0 ns, mínimo;
t_{OTW} (OUTPUT 3-STATE FROM WRITE) - 40 ns, máximo;
t_{DW} (DATA TO WRITE TIME OVERLAP) - 90 ns, mínimo;
t_{OH} (DATA HOLD FROM WRITE TIME) - 0 ns, mínimo.

As memórias RAM dinâmicas exigem um refresh periódico para que os dados não se percam. A figura 11.10 ilustra uma RAM dinâmica 4116 (16 Kbits de capacidade):

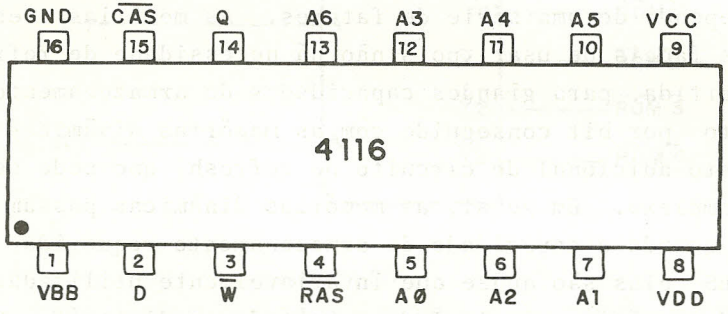


Fig. 11.10 - 4116, 16384 BIT DYNAMIC RAM.

A memória 4116, necessita de três tensões de alimentação: V_{CC} = +5V, V_{DD} = +12V e V_{BB} = - 5V. As linhas A_0 a A_6 são linhas de endereço. Observe que há apenas sete linhas de endereço, embora para endereçar 16 Kbits sejam necessárias quatorze linhas de endereço. A razão disto é que o endereço deve ser apresentado em duas partes. Inicialmente os sete bits menos significativos do endereço são apresentados à memória e o sinal $\overline{RAS}$ (ROW ADDRESS STROBE) é ativado, fazendo com que estes bits sejam armazenados internamente. Em seguida, os sete bits mais significativos do endereço são apresentados à memória e o sinal $\overline{CAS}$ (COLUMN ADDRESS STROBE) é ativado, completando assim os 14 bits necessários ao endereçamento de uma posição de memória. A memória pode ser visualizada como uma matriz de 128 linhas x 128 colunas, onde cada bit corresponde a um elemento da matriz. A linha D é a entrada de dados e Q a saída; quando a memória não está sendo lida, esta saída fica no estado de alta impedância.O

controle das operações de leitura e escrita é feito através do sinal $\overline{W}$: $\overline{W}$ = "1" implica em leitura e $\overline{W}$ = "Ø" implica numa operação de escrita.

O refresh deve ser feito a intervalos de no máximo 2ms e cada fila (128 bits) é reescrita de uma só vez; durante o refresh a saída está em HI-Z.

O tempo de acesso é de 375 ns a 410 ns, conforme a versão da memória. A dissipação de potência é de 462 mW quando a memória está em operação, e apenas 20 mW quando ela não está sendo usada.

A escolha entre usar memórias RAM estáticas ou dinâmicas depende de uma série de fatores. As memórias estáticas são mais fáceis de usar, pois não há necessidade de refresh. Em contrapartida, para grandes capacidades de armazenamento o menor custo por bit conseguido com as memórias dinâmicas compensa o custo adicional do circuito de refresh, que pode ser bastante complexo. Em geral, as memórias dinâmicas passam a ser viáveis quando a capacidade de armazenamento requerida excede 32 KBYTES; elas são quase que invariavelmente utilizadas na construção da memória principal de computadores digitais. Quando a capacidade de armazenamento requerida é inferior a 32KBYTES, as memórias estáticas são a melhor alternativa.

11.5 - Aplicações

Evidentemente, a aplicação primária de uma memória é armazenar informações. No caso das memórias ROM a informação é fixa e somente pode ser lida e, no caso das memórias RAM, a informação pode ser lida e escrita. De maneira geral, as informações são lidas e/ou escritas na memória por um elemento de controle que chamaremos processador. O processador fornece o endereço e os sinais de controle para a memória e, no caso de uma operação de escrita, também os dados; numa operação de leitura, o processador recebe a informação fornecida pela memória. Para que possamos discutir de maneira concreta a utilização das memórias semicondutoras, vamos considerar que a memória está conectada a um processador conforme ilustra a figura 11.11:

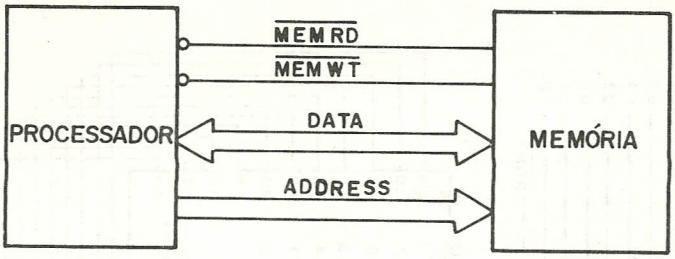


Fig. 11.11 - Conexão processador/memória.

Arbitrariamente vamos assumir que o BUS de endereços consiste de 16 linhas $A_{\emptyset}$ - A_{15}, o BUS de dados bidirecional consiste de oito linhas $D_{\emptyset}$ - D_7 e os sinais de controle $\overline{\text{MEMRD}}$ e $\overline{\text{MEMWT}}$ são ativos no estado lógico "Ø" e indicam, respectivamente, uma operação de leitura e uma operação de escrita. Não nos interessa discutir a construção do processador, mas podemos afirmar que a configuração proposta reflete uma situação real e bastante comum na prática, qual seja o uso de um microprocessador de 8 bits, como o 8085, por exemplo.

Com 16 linhas de endereço é possível endereçar 64K posições de memória e contendo cada posição 8 bits, ou um BYTE, dizemos que a memória pode ter até 64 KBYTES. Os endereços variam entre ØØØØ e FFFF, em hexadecimal.

Vamos supor que se deseje colocar 8KBYTES de memória ROM no sistema da figura 11.11, ocupando os primeiros 8K do espaço de endereçamento e que desejemos utilizar CI's 2716 (2Kx 8). Como cada 2716 contém 2 KBYTES, são necessários 4 CI's para armazenar 8 KBYTES . Os CI's serão interligados conforme ilustra a figura 11.12.

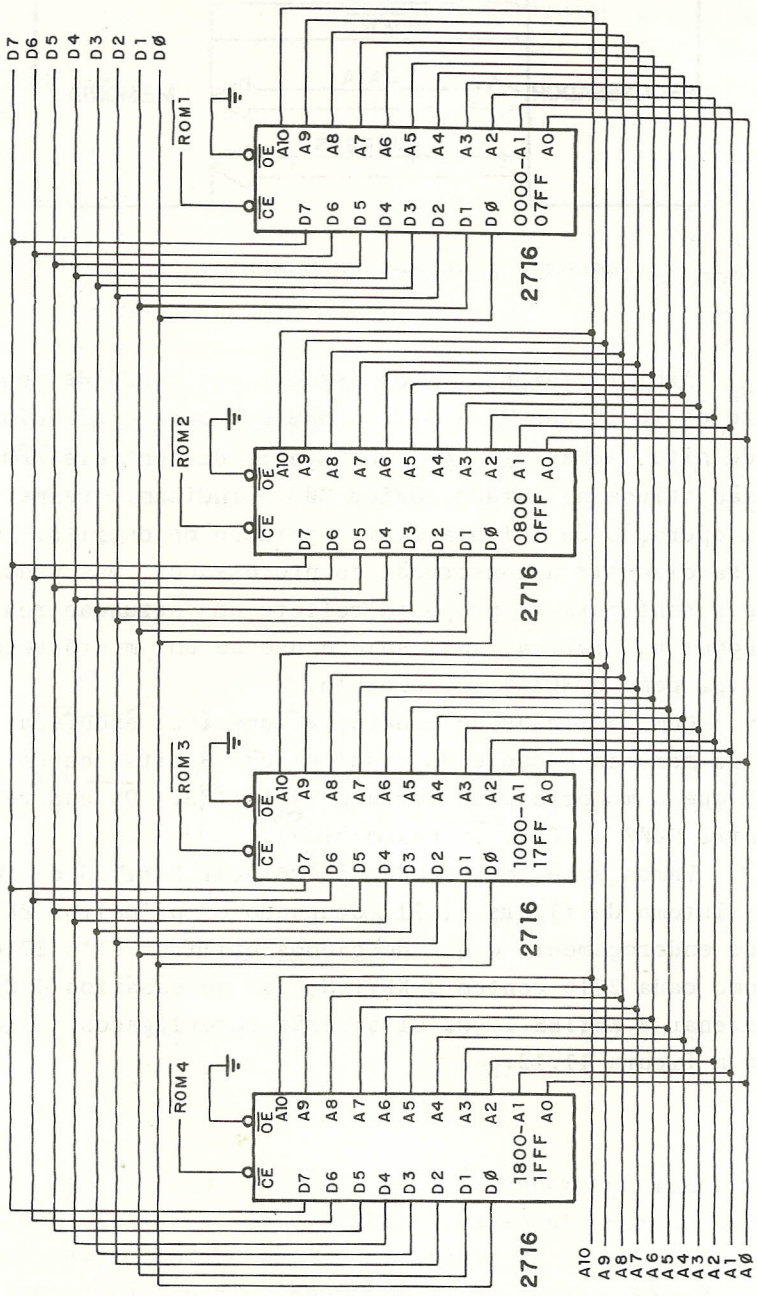


Fig. 11.12 - Ligação de 4 EPROM's 2716 para armazenar 8 KBYTES (endereços de ØØØØ até 1FFF).

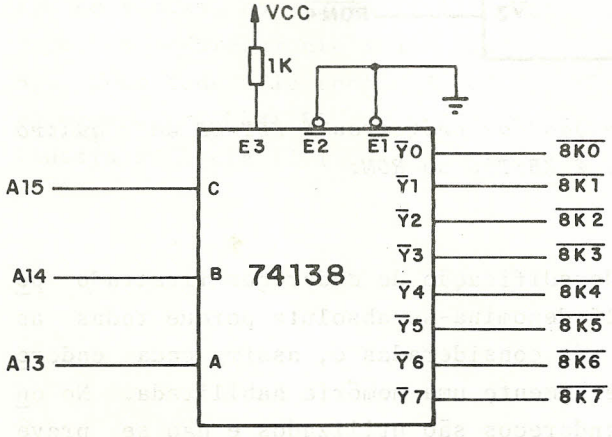
Observe que as saídas $D_\emptyset$ - D_7 são TRI-ESTATE e somente um dos CI's estará habilitado a cada leitura. Vejamos como se pode gerar os sinais de habilitação $\overline{ROM1}$, $\overline{ROM2}$, $\overline{ROM3}$, $\overline{ROM4}$, ativos no estado lógico "Ø".

Uma possibilidade é a decodificação absoluta dos endereços. O espaço de endereçamento de 64K pode ser dividido em 8 faixas de 8K, conforme ilustra a figura 11.13:

ØØØØ	2ØØØ	4ØØØ	6ØØØ	8ØØØ	AØØØ	CØØØ	EØØØ
a	a	a	a	a	a	a	a
1FFF	3FFF	5FFF	7FFF	9FFF	BFFF	DFFF	FFFF

Fig. 11.13 - Divisão do espaço de endereçamento de 64K em 8 faixas de 8K.

Observe que na primeira faixa A_{15} = "Ø", A_{14} = "Ø", A_{13} = "Ø" na segunda faixa A_{15} = "Ø", A_{14} = "Ø", A_{13} = "1" e assim sucessivamente, até que na última faixa A_{15} = "1", A_{14} = "1" e A_{13} = "1". Então, os sinais que indicam em qual das faixas de 8K se enquadra o endereço podem ser gerados utilizando um decodificador de 3 linhas para 8 linhas, conforme ilustrado na figura 11.14:



	ENDEREÇO
$\overline{8K0}$	ØØØØ - 1FFF
$\overline{8K1}$	2ØØØ - 3FFF
$\overline{8K2}$	4ØØØ - 5FFF
$\overline{8K3}$	6ØØØ - 7FFF
$\overline{8K4}$	8ØØØ - 9FFF
$\overline{8K5}$	AØØØ - BFFF
$\overline{8K6}$	CØØØ - DFFF
$\overline{8K7}$	EØØØ - FFFF

Fig. 11.14 - Decodificação do espaço de endereçamento em 8 faixas de 8 KBYTES.

Como desejamos utilizar CI's 2716 para armazenar os primeiros 8 KBYTES, devemos decodificar adicionalmente as linhas de endereço para selecionar um dos quatro CI's. Isto pode ser feito utilizando um decodificador de 2 linhas para 4 linhas, cujas entradas são as linhas de endereçamento A_{12} e A_{11}. Um detalhe adicional deve ser considerado quando se está decodificando o endereçamento de uma ROM. Conforme sabemos, uma ROM pode ser apenas lida e se, por um erro, o processador tentasse executar uma operação de escrita para um endereço de ROM ocorreria a situação em que ambos, o processador e a memória, estariam colocando informações nas linhas de dados. Esta situação jamais deve ocorrer e assim deve-se tomar o cuidado de somente permitir a habilitação da memória ROM se o endereço corresponder a ela e não se tratar de uma operação de escrita. O circuito resultante, que permite selecionar uma das memórias 2716, está ilustrado na figura 11.15:

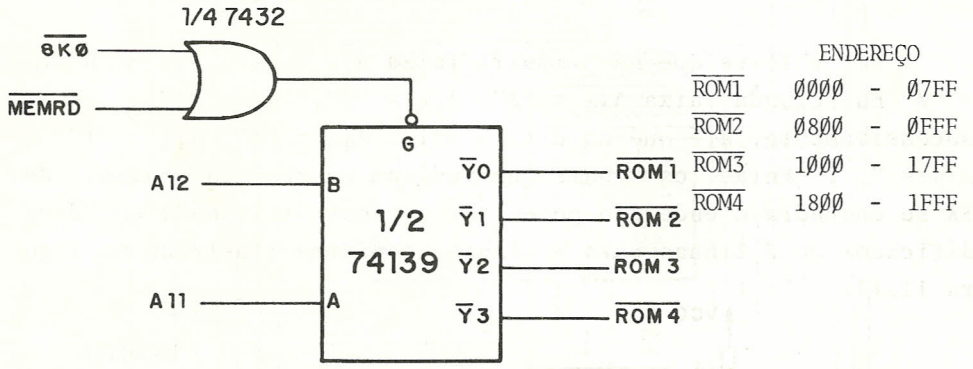
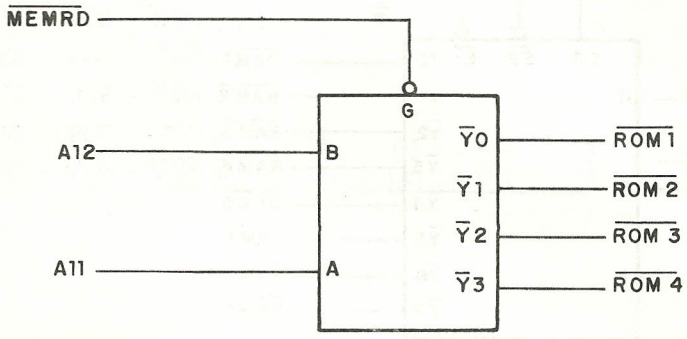


Fig. 11.15 - Decodificação dos primeiros 8 KBYTES em quatro faixas de 2 KBYTES de ROM.

Este tipo de decodificação de endereços ilustrado pelas figuras 11.14 e 11.15 denomina-se absoluta porque todas as linhas de endereçamento são consideradas e, assim, cada endereço corresponderá a uma e somente uma memória habilitada. No entanto, se nem todos os endereços são utilizados e não se prevê expansão da memória pode-se utilizar a decodificação não-absolu

ta. Vamos supor que no espaço de endereçamento de 64 KBYTES só iremos utilizar os primeiros 8 KBYTES. Neste caso, poderíamos utilizar apenas o circuito ilustrado na figura 11.15, desconsiderando as linhas A_{15}, A_{14} e A_{13}. É claro que neste caso os endereços ØØØØ, 2ØØØ, 4ØØØ, 6ØØØ, 8ØØØ, AØØØ, CØØØ e EØØØ corresponderiam todos ao endereço "Ø" da ROM1 e assim sucessivamente, conforme ilustra a figura 11.16:



	ENDEREÇO		
$\overline{ROM1}$	ØØØØ - Ø7FF	2ØØØ - 27FF	... EØØØ - E7FF
$\overline{ROM2}$	Ø8ØØ - ØFFF	28ØØ - 2FFF	... E8ØØ - EFFF
$\overline{ROM3}$	1ØØØ - 17FF	3ØØØ - 37FF	... FØØØ - F7FF
$\overline{ROM4}$	18ØØ - 1FFF	38ØØ - 3FFF	... F8ØØ - FFFF

Fig. 11.16 - Decodificação não-absoluta para endereçamento de 8 KBYTES de ROM.

A escolha entre a decodificação absoluta e não-absoluta dos endereços depende de vários fatores. Em geral, a decodificação absoluta é preferível, pois permite a expansão da capacidade de armazenamento, se necessário; por outro lado, quando não se prevê expansão, a decodificação não absoluta pode ser utilizada e permite economizar alguns circuitos integrados.

A decodificação de endereços para memórias RAM segue os mesmos princípios. A única diferença é que, neste caso, não é necessário considerar o sinal $\overline{MEMRD}$ na habilitação do decodificador uma vez que as memórias RAM podem ser lidas e escritas. Vamos supor que desejamos colocar 8 KBYTES de RAM em nosso sis

tema, ocupando os endereços de 2ØØØ a 3FFF e utilizando RAM's 2114 (1K x 4). Com a decodificação absoluta dos endereços, devemos utilizar um decodificador de 3 linhas para 8 linhas, conforme ilustra a figura 11.17:

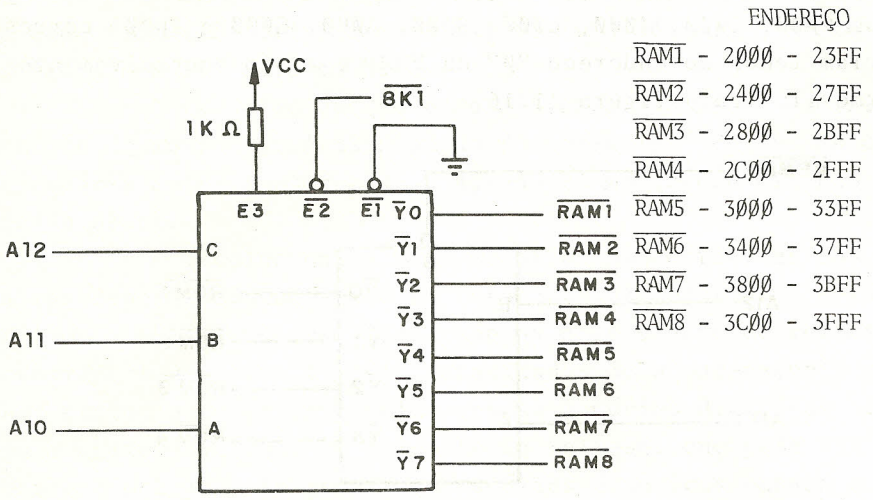


Fig. 11.17 - Decodificação absoluta de endereços para 8 K de RAM.

A ligação das memórias RAM está ilustrada na figura 11.18:

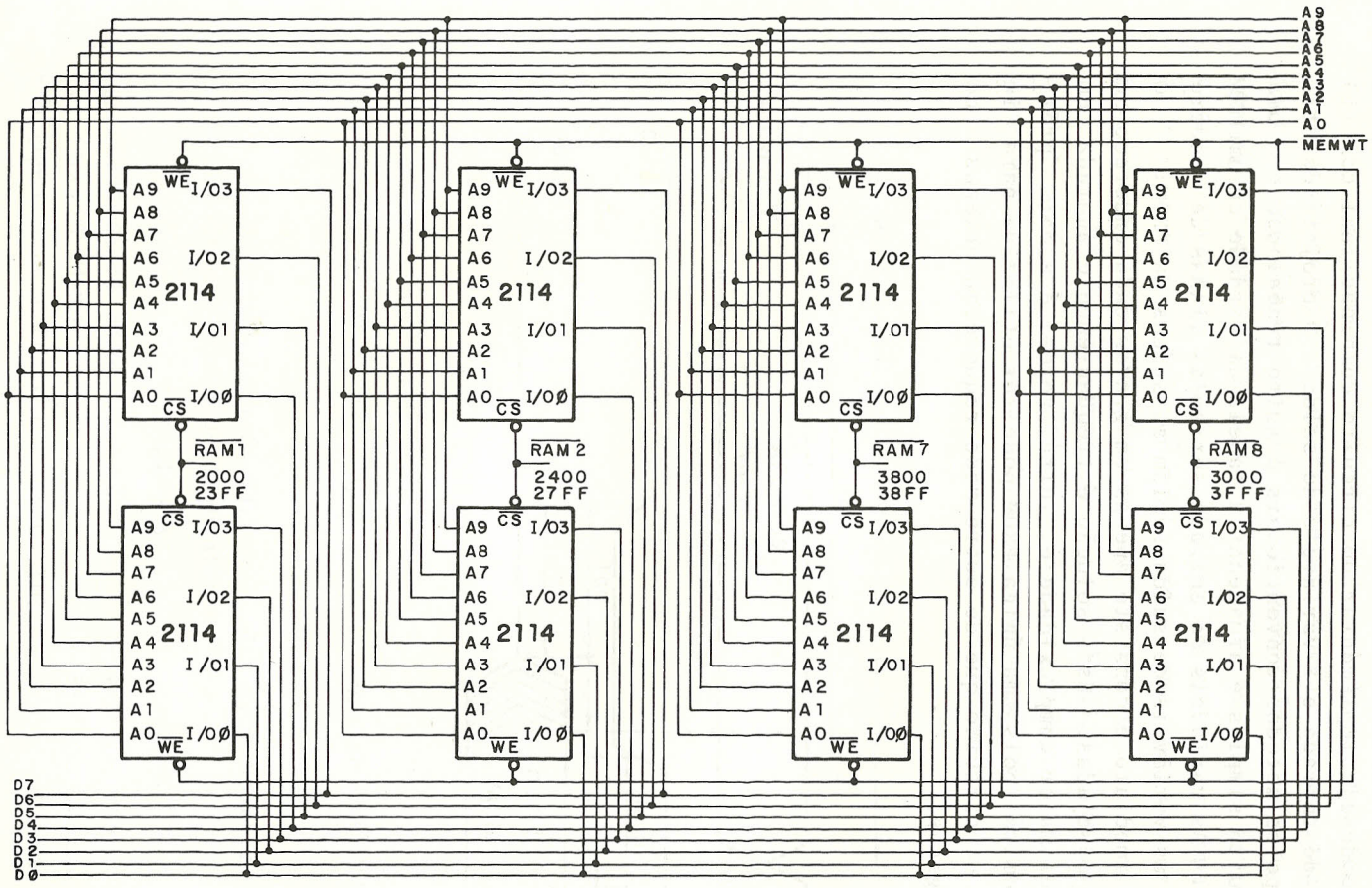


Fig. 11.18 - Ligação de memórias RAM 2114 para armazenar 8 KBYTES (endereços 2000 até 3FFF).

Obs: A figura mostra apenas 8 das memórias ligadas; para as outras a ligação é indêntica.

Uma aplicação interessante das memórias ROM é na implementação de circuitos combinacionais mais complexos. Imaginemos que se deseje implementar um circuito combinacional com 10 variáveis de entrada e 8 variáveis de saída. O projeto de tal circuito pelos métodos convencionais é muito trabalhoso e uma solução muito simples é armazenar a tabela da verdade em uma PROM de 1K x 8. As variáveis de entrada seriam as linhas de endereçamento e as variáveis de saída seriam as saídas da PROM. O projeto é imediato e consiste apenas em gravar cada endereço com as saídas desejadas. Esta solução é freqüentemente adotada em sistemas de maior complexidade e, muitas vezes, é a mais prática e econômica, pois uma única PROM pode ser utilizada para implementar um circuito que de outro modo requereria dezenas de portas lógicas.

CAPÍTULO 12 - CONVERSORES D/A E A/D

12.1 - Introdução

Nos sistemas digitais, ocorre freqüentemente a necessidade de converter-se uma informação digital para analógica (conversão D/A) ou, inversamente, uma informação analógica para digital (conversão A/D).

Como exemplo de emprego do primeiro tipo de conversor, pode-se citar um sistema de comando de um servomecanismo onde a posição de um elemento mecânico é representada internamente de forma binária e comandada diretamente por uma tensão analógica proporcional à informação digital.

O segundo tipo de conversão é bem ilustrado por um sistema de aquisição de dados (DATA ACQUISITION) onde um transdutor converte um parâmetro físico tal como temperatura, pressão ou posição numa tensão ou corrente elétrica. Uma vez em forma elétrica, todo o processamento do sinal é feito por circuitos eletrônicos e, após completado o processamento analogico (amplificação, filtragem, etc), o sinal é convertido numa informação digital e enviado a um computador, controlador digital ou transmissor de dados digitais.

Neste capítulo, serão estudados os circuitos básicos que efetuam a conversão de informações digitais para analógicas e vice-versa.

12.2 - Conceitos Básicos

Antes de passar ao estudo dos conversores D/A e A/D propriamente ditos, é interessante examinar alguns componentes básicos utilizados em sua construção. Tal estudo é necessariamente bastante sucinto e maiores detalhes podem ser encontrados na referência da bibliografia.

Um amplificador operacional ideal é um circuito que apresenta as seguintes características:

- ganho de malha aberta infinito,

- impedância de saída nula,
- impedância de entrada infinita,
- tensão de saída $e_0=0$ quando as tensões nas entradas inversora e não inversora, e_1 e e_2, respectivamente, forem iguais,
- largura de faixa infinita.

A figura 12.1 mostra o símbolo utilizado para representar o amplificador operacional.

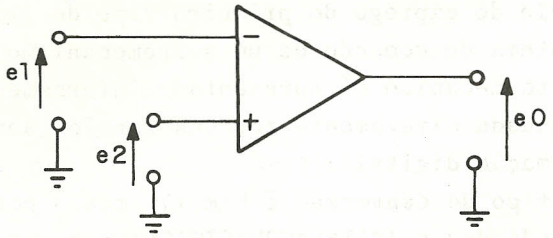


Fig. 12.1 - Símbolo do Amplificador Operacional.

A figura 12.2 mostra o circuito básico de um amplificador inversor utilizando um amplificador operacional:

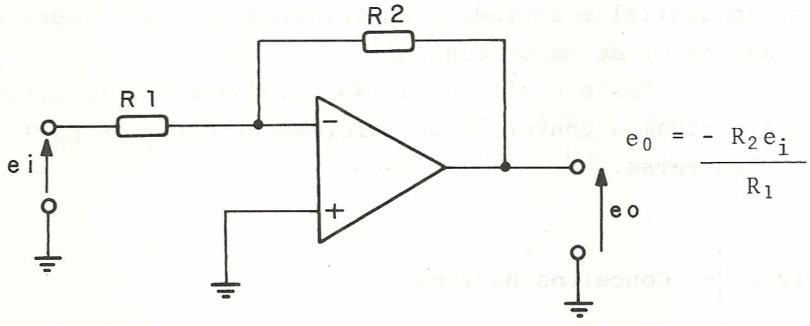
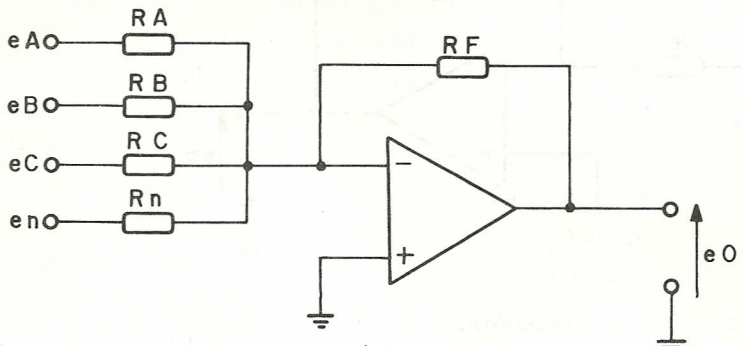


Fig. 12.2 - Amplificador Inversor.

Neste circuito, o ganho é determinado pela relação R_2/R_1, havendo uma inversão de fase de 180° entre a entrada e a saída.

Se houver diversas entradas, a tensão de saída será a soma das tensões produzidas pelas mesmas, consideradas independentemente. Ter-se-á então o chamado somador inversor, cu

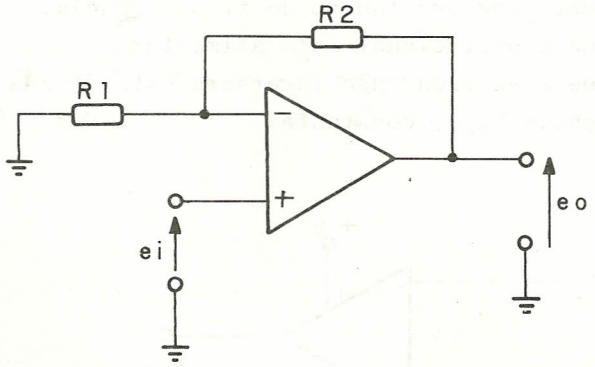
já configuração básica está ilustrada na figura 12.3:



$$e_0 = -R_F\left(\frac{e_A}{R_A} + \frac{e_B}{R_B} + \frac{e_C}{R_C} + \ldots\ldots + \frac{e_n}{R_n}\right)$$

Fig. 12.3 - Somador Inversor Básico.

A configuração do amplificador não-inversor é ilustrada na figura 12.4:



$$e_0 = \left(1+\frac{R_2}{R_1}\right)e_i$$

Fig. 12.4 - Amplificador Não-Inversor.

Se no circuito do amplificador inversor R_2 for substituído por um capacitor, resultará um integrador, assim denominado porque a tensão de saída e_0 é proporcional à integral da tensão de entrada e_i. A figura 12.5 mostra o circuito básico do integrador com o amplificador operacional:

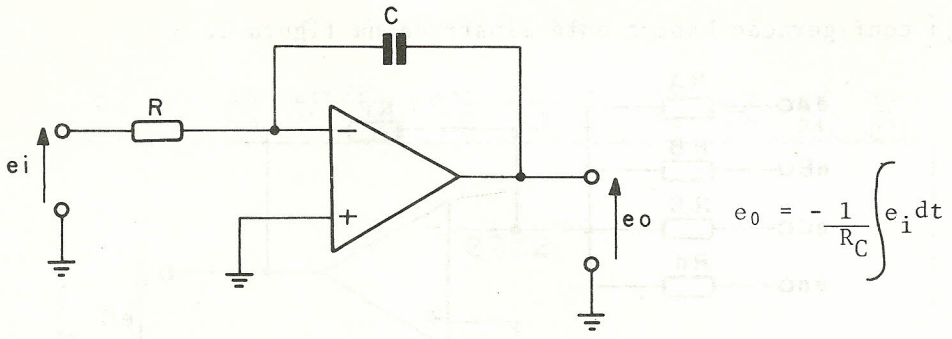
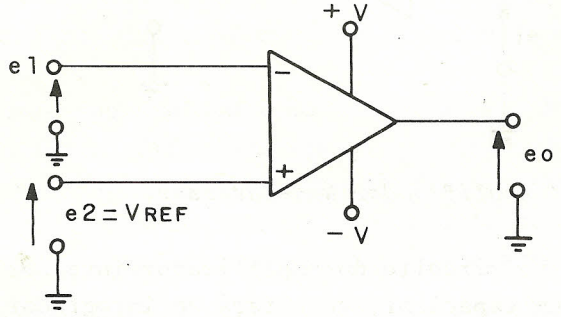


Fig. 12.5 - Integrador.

Como o ganho de malha aberta de um amplificador operacional ideal é infinito, bastará uma diferença infinitesimal entre as tensões de entrada e_1 e e_2 para que a tensão de saída atinja a saturação, ou seja, torne-se igual à tensão de alimentação. Esta característica é utilizada na construção de comparadores, capazes de detetar a grandeza relativa das tensões de entrada e_1 e e_2. Um tipo de circuito comparador é ilustrado na figura 12.6, juntamente com sua função de transferência. Neste circuito supõe-se que o operacional está alimentado com duas tensões +V e -V e que a entrada não inversora está fixada em uma tensão de referência V_{REF}, constante.



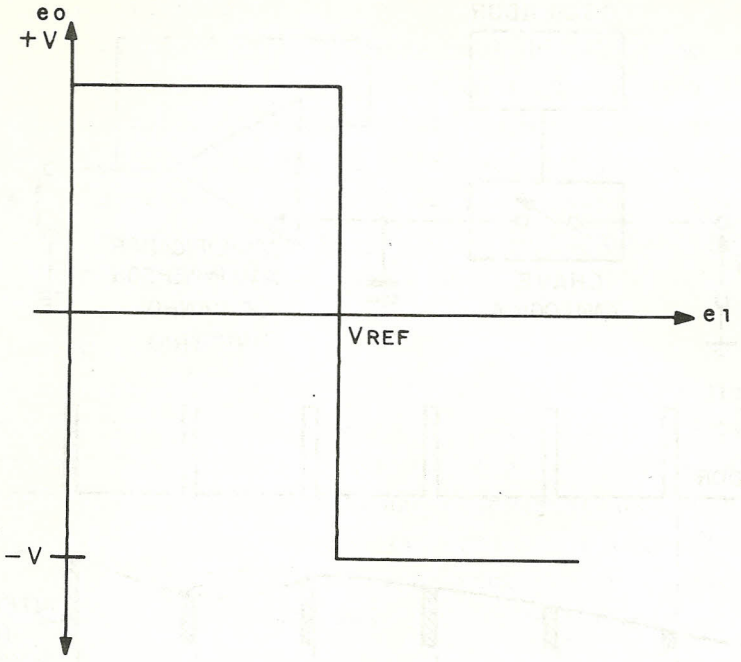


Fig. 12.6 - Comparador Inversor e sua função de transferência

Se a tensão na entrada inversora, e_1, é menor que a tensão de referência, a tensão de saída é igual a +V; se $e_1 = V_{REF}$, $e_0 = \emptyset V$ e, finalmente, se $e_1 > V_{REF}$, $e_0 = -V$.

No processo de conversão de um sinal analógico para digital, muitas vezes é interessante evitar que a tensão analógica a ser convertida varie durante o processo de conversão. Para se conseguir isto, a entrada analógica é amostrada periodicamente e o valor obtido durante a amostragem é armazenado e mantido praticamente constante durante amostragens sucessivas. O circuito que realiza esta função é denominado SAMPLE-AND-HOLD, literalmente amostragem é manutenção. A figura 12.7 ilustra o diagrama básico de um circuito de amostragem e as formas de onda correspondentes:

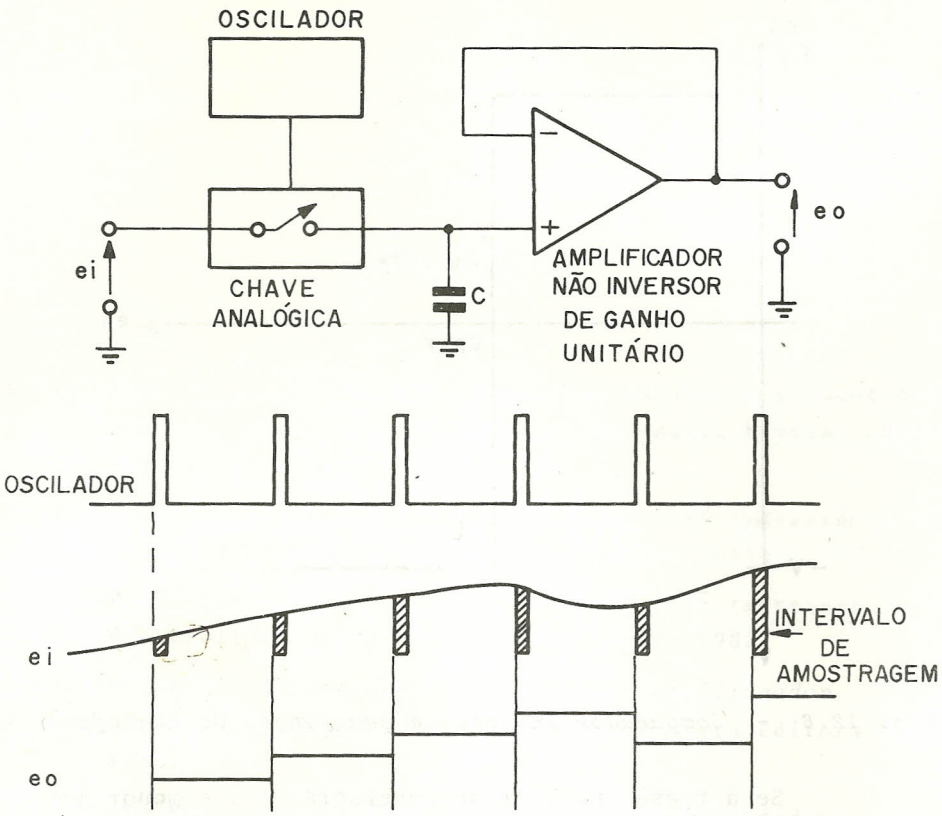


Fig. 12.7 - Circuito de amostragem e formas de onda.

Neste circuito a chave analógica (4066, por exemplo) se fecha durante os intervalos de amostragem, permitindo que o capacitor C se carregue quase instantaneamente com o valor da tensão de entrada e_i. Como a impedância de entrada do amplificador é muito elevada, o capacitor mantém sua carga durante as amostragens sucessivas. É possível demonstrar que o sinal de saída e_0 pode ser reconstituído sem distorção quando a freqüência de amostragem é duas vezes maior que a freqüência da componente mais elevada do mesmo.

É de preferência que a freqüência de amostragem seja maior que duas vezes para uma melhor reconstituição.

12.3 - Conversores Digital-Analógico

A conversão de sinais digitais para analógicos pode ser feita através de circuitos relativamente simples como o ilustrado na figura 12.8, que mostra um conversor D/A para quatro BITS:

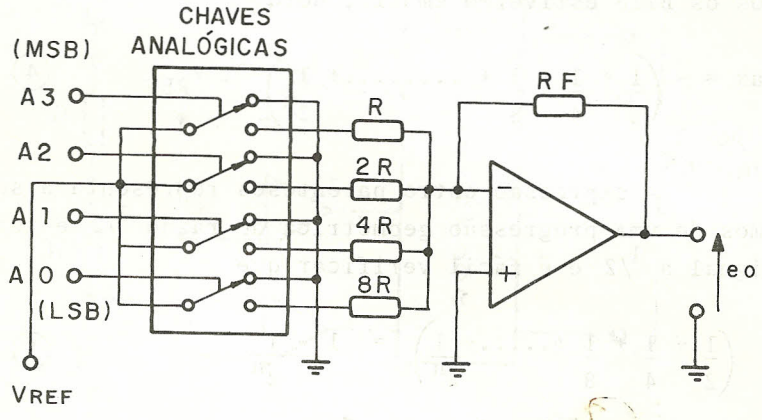


Fig. 12.8 - Conversor D/A com malha de resistores variáveis.

Neste circuito as chaves analógicas são controladas de modo que se o BIT An da entrada digital está no nível lógico "1" a chave correspondente está conectando a tensão V_{REF} à entrada e caso contrário está aterrando a entrada.

A tensão de saída e_0 será dada pela equação:

$$e_0 = - \frac{R_F}{R} \left(A_3 + \frac{A_2}{2} + \frac{A_1}{4} + \frac{A_0}{8} \right) V_{REF} \quad (1)$$

onde os coeficientes A_3, A_2, A_1 e A_0 somente podem assumir os valores "Ø" e "1". Conforme se pode observar, o amplificador operacional é aqui utilizado como um somador inversor.

Este circuito pode ser facilmente estendido para qualquer número de BITS e a equação da saída e_0 para n BITS será:

$$e_0 = - \frac{R_F}{R} \left(An-1 + \frac{1}{2} An-2 + \frac{1}{4} An-3 + \ldots + \frac{1}{2^{n-1}} A_0 \right) V_{REF} \quad (2)$$

que é uma expressão absolutamente geral.

Fazendo $R_F = \frac{1}{2} R$, a expressão acima se simplifica:

$$e_0 = -\left(\frac{1}{2}\,An\text{-}1 + \frac{1}{4}\,An\text{-}2 + \frac{1}{8}\,An\text{-}3 + \ldots + \frac{1}{2^n}\,A_0\right) V_{REF} \qquad (3)$$

O valor mínimo de e_0 é igual a Ø, quando todos os BITS da entrada digital estiverem em Ø; o valor máximo, quando todos os BITS estiverem em "1", será:

$$e_0 max = -\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots\ldots\ldots + \frac{1}{2^n}\right) \cdot V_{REF} \qquad (4)$$

A expressão entre parênteses representa a soma dos n termos de uma progressão geométrica de razão $^1/_2$ e termo inicial igual a $^1/_2$ e é fácil verificar que

$$\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots\ldots + \frac{1}{2^n}\right) = 1 - \frac{1}{2^n} \qquad (5)$$

logo, o valor máximo de e_0 será

$$e_0 max = -\left(1 - \frac{1}{2^n}\right) V_{REF} \qquad (6)$$

nota:

Como o V_{REF}, neste caso, é negativo, temos a expressão final e_0 com valores positivos como ilustra a figura 12.9.

A figura 12.9 mostra a variação da tensão de saída em função da tensão de entrada digital de um conversor D/A de 3 BITS:

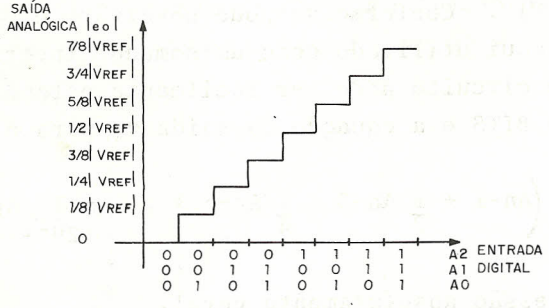
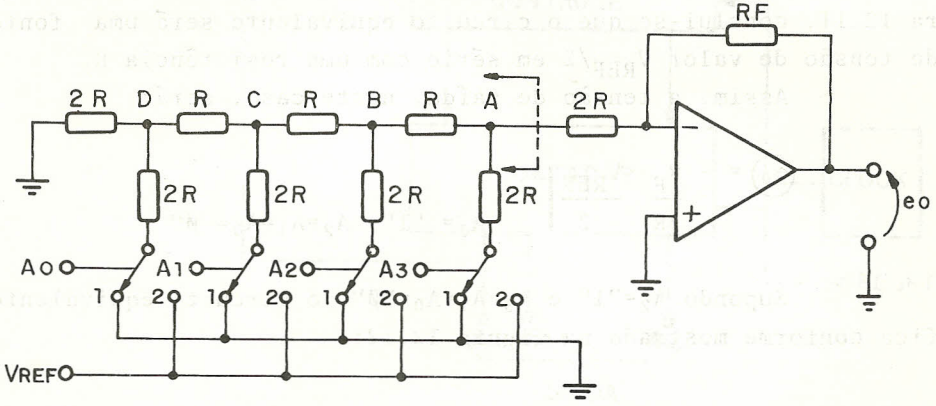


Fig. 12.9 - Tensão de saída e_0 de um conversor D/A de 3 BITS em função das entradas digitais.

O conversor D/A da figura 12.8 apresenta a desvantagem de que a faixa de valores necessários para os resistores de entrada torna-se muito grande à medida que o número de BITS aumenta. Como estes resistores devem ser de precisão, pode ser difícil (e caro) obter todos os valores necessários. Este problema é contornado utilizando na entrada a rede R-2R (BINARY LADDER), que necessita apenas dois valores diferentes de resistores, seja qual for o número de BITS da entrada digital.

A figura 12.10 mostra o circuito de um conversor D/A para quatro BITS utilizando a rede R-2R:



Obs:. Vide nota da equação 6

Fig. 12.10 - Conversor D/A para 4 BITS com rede R-2R.

Neste circuito, se a entrada digital An está no nível lógico "1", a chave analógica correspondente, está na posição 2, caso contrário, está na posição 1.

Para compreender o funcionamento do circuito, considere-se A_3="1" e as demais entradas iguais a "Ø". Neste caso, o circuito equivalente para a parte à esquerda da linha tracejada na figura 12.10 fica como mostrado na figura 12.11:

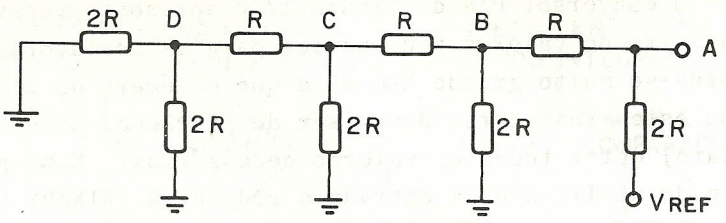


Fig. 12.11 - Circuito equivalente da malha resistiva para A_3 = *"1" e* $A_2=A_1=A_0=$"Ø".

Aplicando o teorema de Thevenin ao ponto "A" na figura 12.11, conclui-se que o circuito equivalente será uma fonte de tensão de valor $V_{REF}/2$ em série com uma resistência R.

Assim, a tensão de saída, neste caso, será:

$$e_0 = -\frac{R_F}{3R}\,\frac{V_{REF}}{2}\Bigg| \qquad A_3 = \text{"1"},\ A_2=A_1=A_0=\text{"Ø"}$$

Supondo A_2="1" e $A_3=A_1=A_0=$"Ø", o circuito equivalente fica conforme mostrado na figura 12.12:

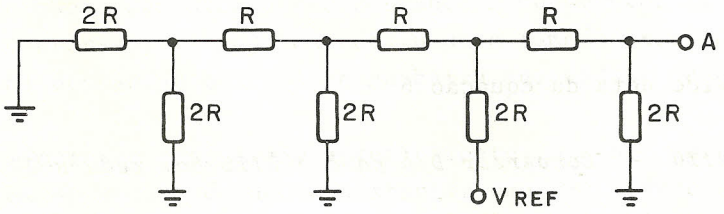


Fig. 12.12 - Circuito equivalente da malha resistiva para $A_2=1$, $A_3=A_1=A_0=$"Ø".

O equivalente de Thevenin para o ponto "A" será uma fonte de tensão de valor $V_{REF}/4$ em série com uma resistência R. A tensão de saída será, neste caso,

$$e_0 = -\frac{R_F}{3R}\,\frac{V_{REF}}{4}\Bigg| \qquad A_3=\text{"Ø"},\ A_2=\text{"1"},\ A_1=A_0=\text{"Ø"}$$

Procedendo de maneira semelhante para os demais BITS conclui-se que:

$$e_0 = -\frac{R_F}{3R}\frac{V_{REF}}{8}\Bigg|_{A_3="\emptyset",\ A_2="\emptyset",\ A_1="1",\ A_0="\emptyset"}$$

$$e_0 = -\frac{R_F}{3R}\frac{V_{REF}}{16}\Bigg|_{A_3="\emptyset",\ A_2="\emptyset",\ A_1="\emptyset",\ A_0="1"}$$

Aplicando o princípio da superposição tem-se a expressão geral da tensão de saída e_0 em função das entradas digitais:

$$e_0 = -\frac{R_F}{3R}\left(\frac{1}{2}A_3 + \frac{1}{4}A_2 + \frac{1}{8}A_1 + \frac{1}{16}A_0\right) V_{REF}$$

para n BITS tem-se:

$$e_0 = -\frac{R_F}{3R}\left(\frac{1}{2}An\text{-}1 + \frac{1}{4}An\text{-}2 + \frac{1}{8}An\text{-}3 + \ldots + \frac{1}{2^n}A_0\right) V_{REF}$$

a qual se simplifica para

$$e_0 = -\left(\frac{1}{2}An\text{-}1 + \frac{1}{4}An\text{-}2 + \frac{1}{8}An\text{-}3 + \ldots + \frac{1}{2^n}A_0\right) V_{REF}$$

quando $R_F = 3R$

12.4 - Conversores Analógico-Digital

Há diversos métodos para conversão de sinais analógicos em digitais, dos quais serão examinados os mais comuns.

A figura 12.13 ilustra um conversor A/D tipo contador de rampa (RAMP COUNTER A/D CONVERTER):

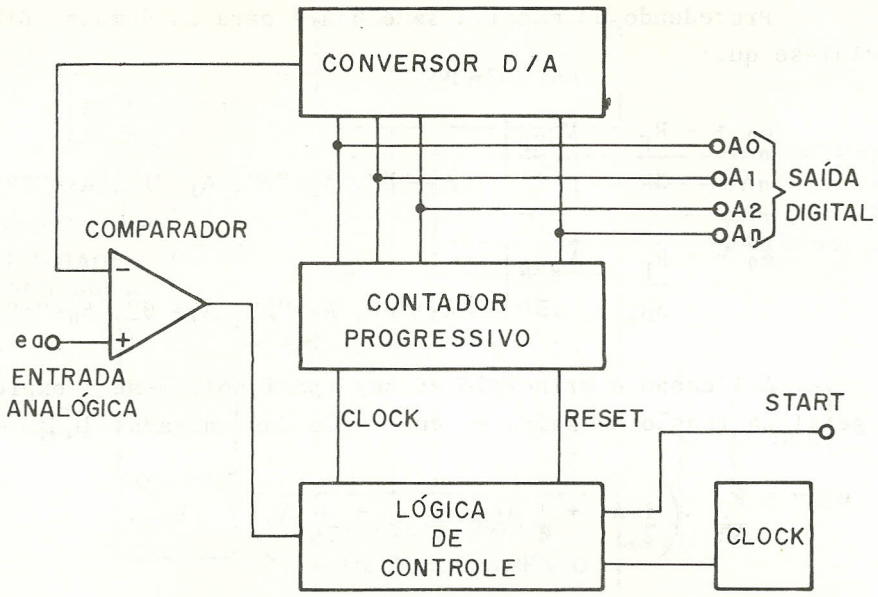


Fig. 12.13 - Conversor A/D contador de rampa.

A conversão tem início com um pulso de START, que provoca o zeramento do contador. A saída do contador é convertida em um nível analógico e comparada com a tensão de entrada. Enquanto a tensão e_a for maior que o valor representado pelas saídas do contador, este recebe os pulsos de CLOCK e vai incrementando sua contagem. No instante em que a tensão e_a fica menor que o valor representado pelas saídas do contador os pulsos de CLOCK para o mesmo são inibidos e a contagem pára. O valor presente na saída digital representa o equivalente da tensão analógica de entrada, com um erro de $\pm 1/2$ LSB.

Este conversor tem como vantagens a simplicidade, baixo custo e boa precisão, mas tem a desvantagem de ser lento. O tempo de conversão é proporcional ao valor da entrada analógica, sendo mais longo para valores mais altos.

Em algumas aplicações onde se quer uma velocidade um pouco maior, usa-se um contador progressivo/regressivo, obtendo-se o chamdo conversor A/D tipo contador de rampa contínuo (CONTINUOUS RAMP COUNTER A/D CONVERTER), cujo diagrama está ilustrado na figura 12.14:

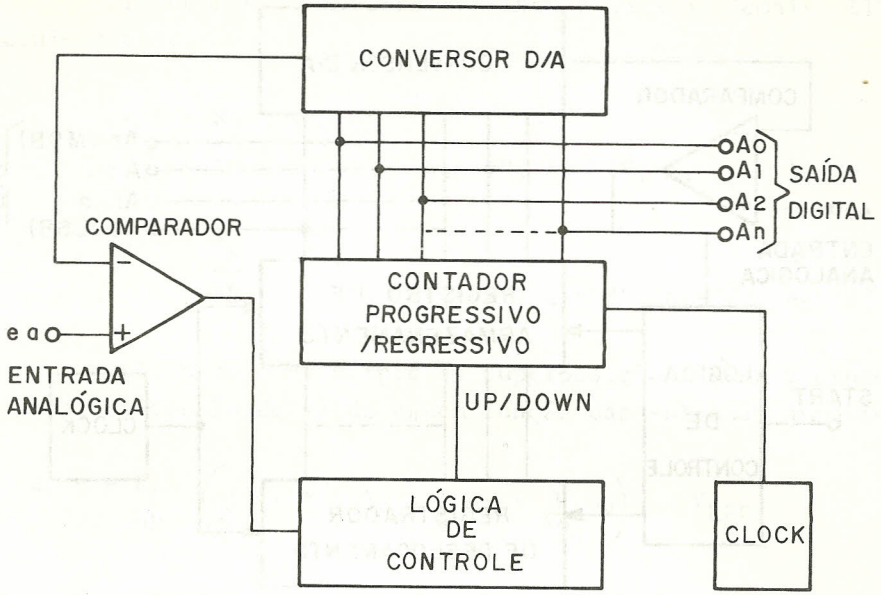


Fig. 12.14 - Conversor A/D tipo contador de rampa contínuo.

Neste tipo de conversor não há necessidade do sinal de START e o contador inicia a contagem partindo do valor anterior. Se a entrada analógica é maior que o valor representado pelas saídas do contador, a contagem é progressiva, caso contrário será regressiva. O conversor está continuamente realizando a conversão e, por isso, a saída fica oscilando em torno do valor final.

Quando a velocidade de conversão é o fator mais importante, o tipo mais utilizado é o conversor A/D de aproximação sucessiva (SUCCESSIVE APROXIMATION A/D CONVERTER), cujo diagrama é apresentado na figura 12.15. Neste conversor, o tempo de conversão é independente do valor da entrada analógica e sempre igual a n pulsos de CLOCK, onde n é o número de BITS.

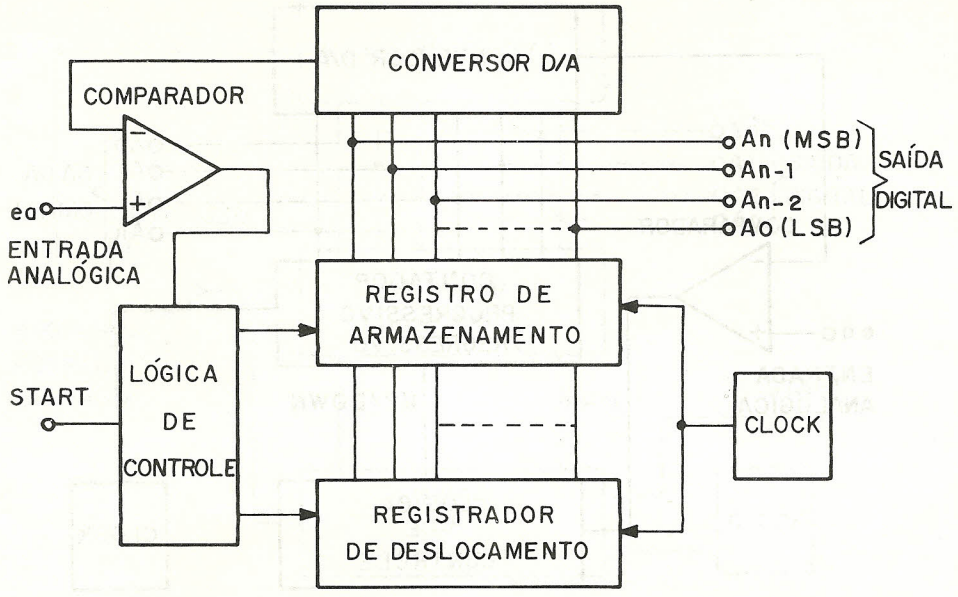


Fig. 12.15 - Conversor A/D de aproximação sucessiva.

Ao iniciar-se o processo de conversão com o pulso de START, o MSB da saída digital é setado. Se a saída digital é menor que a entrada analógica, o MSB permanece em "1" e o BIT seguinte é setado; caso contrário, o MSB é ressetado e o BIT seguinte é setado. O processo de comparação descrito acima continua até o LSB, após o que o registro de armazenamento contém o equivalente digital da saída analógica. Com este processo é possível conseguir velocidade de conversão de até 100 ns/BIT.

Os conversores analógico-digitais vistos até o momento utilizam conversores D/A em sua construção. Um outro tipo de conversor A/D, denominado conversor A/D por integração, dispensa o uso de conversores D/A e opera pelo princípio de converter a tensão analógica de entrada em um intervalo de tempo que é então medido por um contador. O mais utilizado destes conversores denomina-se conversor A/D integrador por dupla inclinação (DUAL SLOPE INTEGRATING A/D CONVERTER) e seu diagrama é ilustrado na figura 12.16:

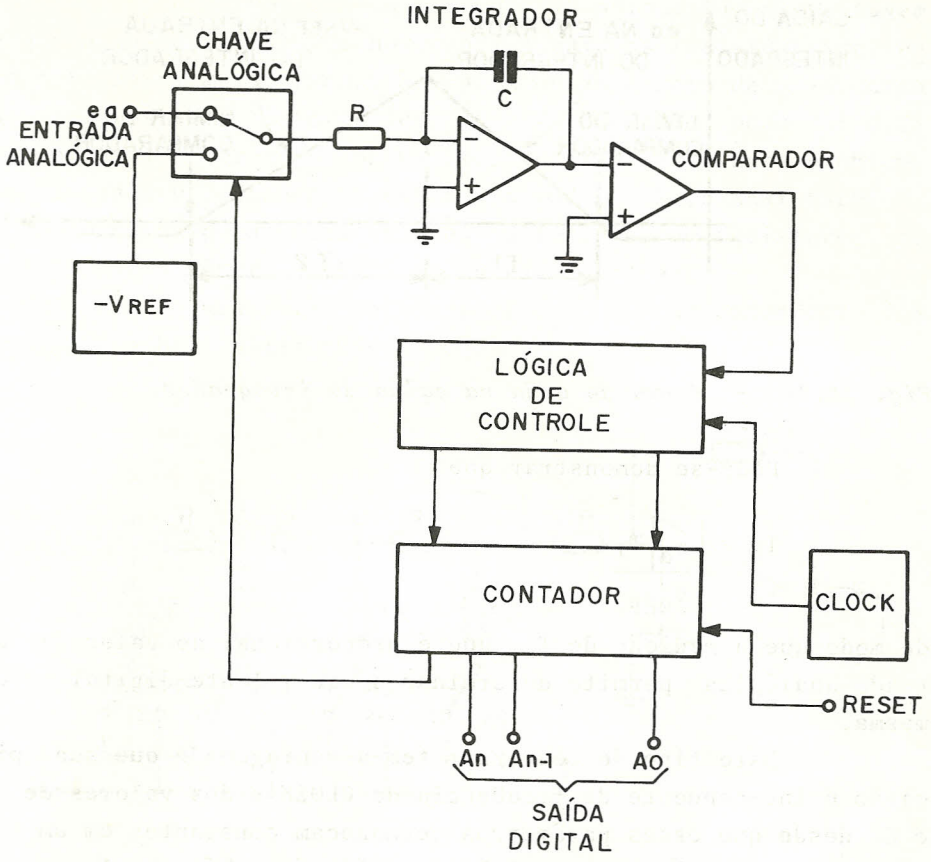


Fig. 12.16 - Conversor A/D integrador por dupla inclinação.

Inicialmente o contador é zerado por um pulso de RESET e o zero no último estágio faz com que a chave analógica coloque a tensão de entrada e_a na entrada do integrador, conforme ilustrado na figura 12.16. A tensão na saída do integrador cresce linearmente e, no instante em que o limiar do comparador é cruzado, pulsos de CLOCK são aplicados ao contador. A saída do contador é incrementada e quando se torna igual a 1ØØØ...Ø a chave analógica comuta, aplicando agora a tensão negativa de referência à entrada do integrador. A tensão na saída do integrador passa agora a decrescer linearmente, até que o limiar do comparador seja novamente cruzado, quando os pulsos de CLOCK para o contador são interrompidos. A figura 12.17 mostra a forma de onda na saída do integrador:

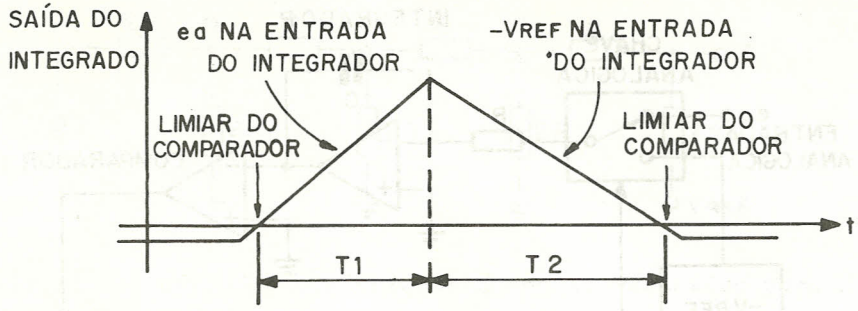


Fig. 12.17 - Forma de onda na saída do integrador.

Pode-se demonstrar que

$$T_2 = \left| \frac{e_a}{V_{REF}} \right| T_1$$

de modo que a medição de T_2, que é proporcional ao valor da entrada analógica, permite determinar o equivalente digital da mesma.

Este tipo de conversor tem a vantagem de que sua precisão é independente da freqüência do CLOCK e dos valores de R e C, desde que esses parâmetros permaneçam constantes em um período de conversão. A rejeição de ruído é também excelente e a principal desvantagem consiste na baixa velocidade de conversão.

Conforme ressaltado anteriormente, o conversor A/D de aproximação sucessiva é capaz de atingir uma velocidade de 1ØØns/BIT resultando numa taxa de conversão de 1 MHZ para 10 BITS. No entanto, o mais rápido processo de conversão, denominado conversão paralela, permite atingir taxas de até 25 MHZ para 4BITS. A figura 12.18 ilustra um conversor A/D paralelo:

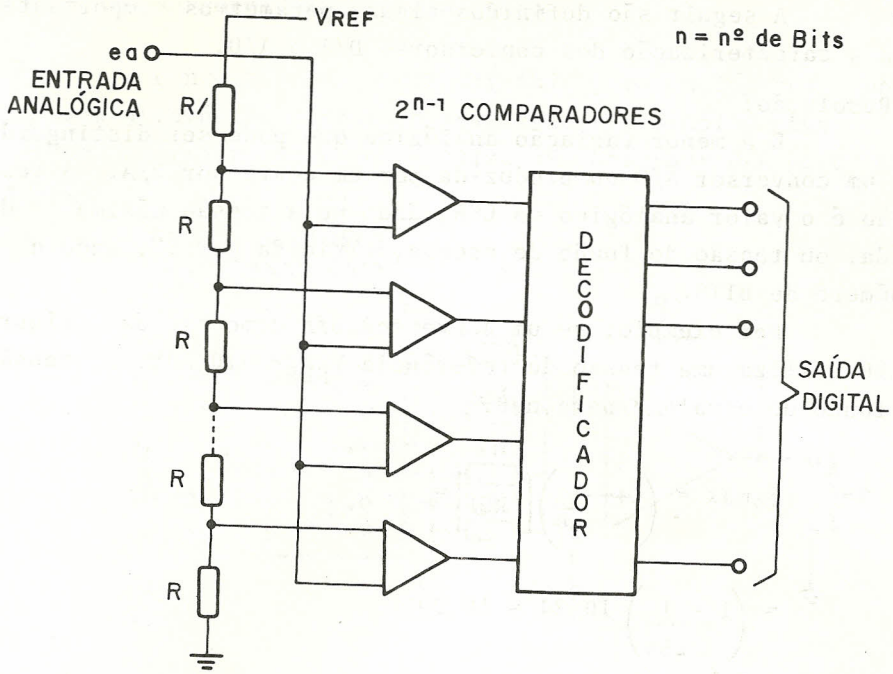


Fig. 12.18 - Conversor A/D paralelo.

Como se vê pelo esquema, neste conversor há 2^{n-1} comparadores polarizados com uma diferença de 1 LSB entre si. Para um dado valor de e_a todos os comparadores abaixo deste valor estão em um estado e todos os comparadores acima estão em outro estado. As saídas dos comparadores vão a um decodificador, pois não estão em código binário. O conversor A/D paralelo tem a vantagem de altíssima velocidade, mas seu uso está limitado a um número pequeno de BITS, (em geral um máximo de 4) devido ao grande número de comparadores requerido.

12.5 - Parâmetros dos Conversores D/A e A/D

A seguir são definidos alguns parâmetros importantes para a caracterização dos conversores D/A e A/D.

a) Resolução:

É a menor variação analógica que pode ser distinguida por um conversor A/D ou produzida por um conversor D/A. A resolução é o valor analógico do LSB, dado pela tensão máxima de saída, ou tensão de fundo de escala, dividida por 2^n, onde n e o número de BITS.

Por exemplo, se um conversor D/A como o da figura 12.10 utiliza uma tensão de referência $V_{REF} = -10,24V$, a tensão de fundo de escala é para n=8.

$$e_0 max = \left(1 - \frac{1}{2^n}\right)\left|V_{REF}\right| =$$

$$= \left(1 - \frac{1}{256}\right) 10,24 = 10,20V$$

e a resolução é

$$R = \frac{e_0 max}{2^n} = \frac{10,20}{256} = 40\ mV$$

b) Linearidade:

A figura 12.19 ilustra a curva de transferênica ideal de um conversor A/D. A linearidade é definida como o desvio máximo em relação à linha reta traçada entre os pontos extremos da função de transferência, conforme ilustrado na figura. Este parâmetro é geralmente expresso como uma fração do valor do LSB e para um bom conversor deve ser de no máximo $\pm 1/2$ LSB:

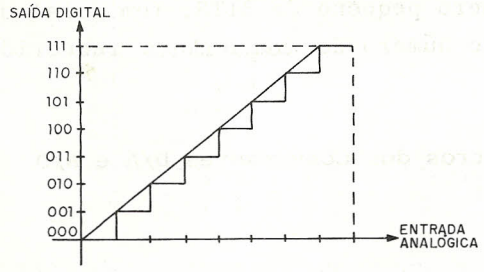


Fig. 12.19 - Curva de transferência de um conversor A/D de 3 BITS.

c) Linearidade Diferencial

É o desvio máximo do valor real de um BIT em relação a seu valor teórico para qualquer BIT em toda a escala do conversor. Por exemplo, calculou-se anteriormente que, para o conversor D/A citado no item a), os BITS seriam espaçados de 40 mV. Na prática, alguns podem ter 30 mV ou 50 mV, por exemplo. Se a linearidade diferencial é de $\pm \frac{1}{2}$ LSB isto significa que cada BIT pode estar espaçado 1 LSB $\pm \frac{1}{2}$ LSB.

d) Monotonicidade:

A monotonicidade significa ter uma saída regular e continuamente crescente para uma entrada regular e continuamente crescente. Isto implica em que um "degrau" correspondente a um nível de entrada mais elevado não pode estar abaixo de outro correspondente a um nível de entrada menor (ver fig. 12.19). Para que a curva de transferência seja monotônica é necessário que a linearidade diferencial seja menor que 1 LSB.

e) Erro de Quantização:

Em um conversor A/D há um número infinito de valores possíveis para a tensão de entrada, mas somente 2^n códigos de saída. Assim, haverá um erro inerente ao processo de quantização que deve ser no máximo $\frac{1}{2}$ LSB e que será máximo quando a entrada tiver justamente um valor correspondente à transição entre dois níveis de saída.

f) Velocidade de Conversão:

É a medida que indica o tempo necessário para que o conversor forneça uma saída válida. Em geral, é medido entre o pulso de início de conversão e sinal que indica o fim do processo.

18
BV721416594-1

BIBLIOGRAFIA

Livros:

IDOETA, Ivan Valeije e CAPUANO, Francisco Gabriel - *Elementos de Eletrônica Digital*, 6ª Edição, São Paulo, Livros Érica Editora Ltda, 1983.

LANCASTER, Donald - *CMOS COOKBOOK*, 1ª Edição, 6ª Impressão, Indianapolis, Howard W. Sams & Cv.,Inc., 1981.

LANCASTER, Donald - *TTL* , 1ª Edição, 12ª Impressão, Indianapolis, Howard W. Sams & Cv.,Inc., 1982.

LANDO, Roberto Antonio e ALVES, Serg Rios - *Amplificador Operacional*, 1ª Edição, São Paulo, Livros Érica Editora Ltda, 1983.

TEXAS INSTRUMENTS INC. - *Designing With TTL Integrated Circuits*, New York, McGraw-Hill Book Company, 1971.

Manuais:

CMOS Integrated Circuits - Harris Semiconductor.

System 74 Designer's Manual - Texas Instruments Inc.

The TTL Data Book - Texas Instruments Inc.

Impressão e acabamento
(com filmes fornecidos):
EDITORA SANTUÁRIO
Fone (0125) 36-2140
APARECIDA - SP

OUTRAS PUBLICAÇÕES

- □ INFORMÁTICA
- □ ELETRÔNICA
- □ TELECOMUNICAÇÕES
- □ PROCESSAMENTO DE DADOS
- □ MICROPROCESSADORES
- □ ELETROTÉCNICA
- □ MECÂNICA
- □ CIÊNCIAS EXATAS

ÉRICA

LIVROS DE CONTEÚDO, COM QUALIDADE
RUA JARINÚ, 594 - TATUAPÉ - CEP 03306
CX. P. 15617 - TEL.: (011) 294-8686 - S.P.